MLRec 2015

1st International Workshop on Machine Learning Methods for Recommender Systems

May 5, 2015, Vancouver, British Columbia, Canada

In Conjunction with
15th SIAM International Conference on Data Mining (SDM 2015)

Workshop Chairs

George Karypis, University of Minnesota
Jiayu Zhou, Samsung Research America
Deguang Kong, Samsung Research America

Publicity Chair

Shiyu Chang, University of Illinois at UrbanaChampaign

Program Committee

Shengbo Guo, Yahoo! Labs
Xia Ning, Indiana University - Purdure University Indianapolis
Yong Ge, University of North Carolina at Charlotte
Mohit Sharma, University of Minnesota
Lei Cen, Purdue University
Bing Hu, Samsung Research America
Yin Zhou, Samsung Research America
Jianpeng Xu, Michigan State University

Invited Speakers

Chih-Jen Lin, National Taiwan University
Martin Ester, Simon Fraser University
Jieping Ye, University of Michigan
Fei Wang, University of Connecticut
Sofus Macskassy, Facebook
Junling Hu, Samsung Research

Overview

The MLRec 2015 workshop aims to bring the attention of researchers to the various data mining and machine learning methods for recommender systems.

Since the introduction of recommender system, there are a lot of machine learning and data mining algorithms designed for effective and efficient recommendation. To name a few, the matrix factorization techniques are widely used to model the latent space in which users and items interact with each other. The factorization machine uses bilinear regression models to capture the non-linear interactions among the user features and item features. In the past years, researchers have utilized many machine learning techniques such as online learning, metric learning, sparse learning, multi-task learning also to foster the development of recommender systems.

This workshop focuses on applying novel as well as existing machine learning and data mining methodologies for improving recommender systems. Indeed there are many established conferences such as NIPS and ICML that focus on the study of theoretical properties of machine learning algorithms. On the other hand, the recent developed conference ACM RecSys focuses on different aspects of designing and implementing recommender systems. We believe that there is a gap between these two ends, and this workshop aims at bridging the recent advances of machine learning and data mining algorithms to improving recommender systems. Since many recommendation approaches are built upon data mining and machine learning algorithms, these approaches are deeply rooted in their foundations. As such, there is an urgent need for researchers from the two communities to jointly work on 1) what are the recent developed machine learning and data mining techniques that can be leveraged to address challenges in recommender systems, and 2) from challenges in recommender systems, what are the practical research directions in the machine learning and data mining community.

We encourage submissions on a variety of topics, including but not limited to:

1. Novel machine learning algorithms for recommender systems, e.g., new content aware recommendation algorithms, new algorithms for matrix factorization handling cold-start items.

2. Novel approaches for applying existing machine learning algorithms, e.g., applying bilinear models, sparse learning, metric learning, neural networks and deep learning, for recommender systems.

3. Novel optimization algorithms and analysis for improving recommender systems, e.g., parallel/distributed optimization techniques and efficient stochastic gradient descent.

4. Industrial practices and implementations of recommendation systems, e.g., feature engineering, model ensemble, and lessons from large-scale implementations of recommender systems.

We believe that advancements on these topics will benefit a variety of algorithm and application domains.

Acknowledgments

We appreciate the efforts of the Program Committee for their comments and feedback on the submissions.

Table of Contents

Similarity Dependency Dirichlet Process for Aspect-Based Sentiment Analysis

Wanying Ding* Xiaoli Song† Yue Shang‡ Junhuan Zhu§ Lifan Guo¶
Xiaohua Hu‖

Abstract

Aspect-base Sentiment Analysis is a core component in *Review Recommendation System*. With the booming of customers' reviews online, an efficient sentiment analysis algorithm will substantially enhance a review recommendation system's performance, providing users with more helpful and informative reviews. Recently, two kinds of LDA derived models, namely *Word Model* and *Phrase Model*, take the dominant positions in this field. However, the requirement of exact aspect number underlies the usability and flexibility of these LDA extended models. Although, Dirichlet Process(DP), which can help to automatically generate the number of aspects, has been on trial, its random word assignment mechanism makes the result unsatisfying. This paper proposes a model named Similarity Dependency Dirichlet Process(SDDP) to cope with the above problems. SDDP inherits the merits of DP to automatically determine the number of aspects, but it exploits the semantic similarities between words to infer aspects and sentiments, alleviating the random assignment problem in DP. Furthermore, based on SDDP, this paper builds a word model W-SDDP and a phrase model P-SDDP respectively to detect aspects and sentiments from two different perspectives. Finally we experiment both our two models on 6 datasets, and compare with other 6 currently most popular models. The result shows that both W-SDDP and P-SDDP outperform the other 6 models, indicating SDDP is a promising model for sentiment analysis.

1 Introduction

Social media has provided an open platform for users to exchange ideas online, and more and more valuable opinions have been published on different websites,like Amazon and Yelp. But with the dramatic increase in volume, it will cost customers dozens of hours going through all the reviews. Thus, *Review Recommendation System* has been created in order to present customers with most helpful or informative reviews. Within such a system, aspect-based sentiment analysis is an important component[1],[2] to make it function, because it can efficiently detect the ***Aspect*** *(A particular feature, like food or service of a restaurant)* and corresponding ***Sentiments*** *(The subjectivity polarities, like positive or negative)*,and further provide vital features to recognize helpful reviews.

As a core component of recommendation system, sentiment analysis has been long explored. In the early days, classification methods were widely applied [3],[4],[5],[6], and laid a solid foundation. However, the requirement of training dataset presents a challenge for related researches, since manually constructing training datasets is time consuming and laborious, but more and more online data presents as unlabelled. Thus, unsupervised methods become more attractive because of its label-free and flexible in use. The birth of Latent Dirichlet Allocation(LDA)[7] has inspired a lot of researchers in developing unsupervised models[8],[9],[10],[11],[12],[13],[14]. LDA extended models inherit both LDAs advantages and disadvantages. Speaking of advantages, they are all training dataset free, and efficient in aspect and sentiment discovery. The disadvantage is that they require users to decide the number of aspects beforehand. Such decisions are hard to make, especially when one has little knowledge about the dataset at hand. Although it is possible to repeatedly check the perplexity values to find an optimal number of aspects, it has a major practical limitation in that one may have to make a large number of trials. Recently, a few researchers try to replace the static Dirichlet allocation in LDA with dynamic Dirichlet process (DP), which can automatically generate the aspect number according datasets' own characteristics. Jianfeng Si[15] took the first try in implementing DP to discover the sentiment from Twitter, and Kim Suin[16]adopted the Hierarchical DP(HDP)[17] to explore the hierarchical aspect structure from online reviews. Both Jianfeng Si and Kim Suin's works are just simply to apply traditional DP, and ignore the random word assignment problem, which is emphasized by Blei

*College of Computing and Informatics,Drexel University.
†College of Computing and Informatics,Drexel University.
‡College of Computing and Informatics,Drexel University.
§Department of Computer Science, Rochester University.
¶TCL Research America.
‖College of Computing and Informatics,Drexel University.

and Frazier[18].

Considering all the problems mentioned above, this paper has proposed a Similarity Dependency Dirichlet Process (SDDP) for sentiment analysis. Comparing to LDA, SDDP can automatically determine the number of aspects, saving users from laborious trials. Comparing to DP or HDP, SDDP replaces the random assignment mechanism with semantic similarity assignment, making the model more reasonable and acceptable. Besides, considering the *Word Model* and the *Phrase Model* are two research logistics in sentiment analysis area, this paper builds two models based on SDDP, one word model (W-SDDP) and one phrase model (P-SDDP), to recognize one review's sentiment from two different perspectives. We experiment both W-SDDP and P-SDDP on 6 datasets, and compare the results with 6 other models. The results show that both W-SDDP and P-SDDP outperform other models, and this indicates that SDDP is a promising model for use in sentiment analysis.

2 Related Work

Aspect-based sentiment analysis has been deeply explored in the past decades, and most of them exploit Latent Dirichlet Allocation (LDA). Based on LDA, the word model and the phrase model are two research branches that have been drived[19].

Phrase Model processes the raw text data into a series of phrases, which can be shown as $< head\ word, modifier\ word >$. Generally speaking, the head word is used for identifying aspects, and the modifier word is for identifying sentiments. Several tries[12],[20] have been done, but because of the laborious data pre-process work, most researchers prefer the word model.

Word Model relies on the "bag-of-word" assumption. Two methods are commonly used to deal with words. First, some researches assume one word simultaneously conveys both sentiment and aspect information, so they use different aspect-sentiment prior distributions to infer the word assignment. Models like JST[8] and ASUM[21] belong to this category, which is referred as *Pure Word Model* in this paper. Second, external knowledge, like POS tagging or sentiment dictionary, might be used to help distinguish whether a word conveys aspect or sentiment information, and use aspect words to infer the aspects, and sentiment words to infer the sentiments. Models like JAS[22] and MaxEnt-LDA[10] fall into this category. Comparing to the former category, these models is closer to phrase model, since it also needs much data pre-process work. The only difference is that it does not need to pair up the aspect and sentiment words. This paper refers this kind of word model as *Mixture Word Model*.

Currently, most models are LDA based. The problem of LDA extended models is that the number of aspects needs to be pre-determined. However, such a decision is hard to make, especially in the social media environment. For example, one can hardly know how many aspects people may mention about towards a restaurant in Yelp.com. Traditionally, researchers will take a lot of trials and errors to find the optimal number, but it is time and labor consuming. One solution to this problem is to replace the Dirichlet allocation in LDA with Dirichlet process, which can help to generate the number of aspects automatically. Thus, Hierarchical Dirichlet Process(HDP)[17] was introduced as a non-parametric model to solve the aspect number determination problem.

HDP consists of two levels of Dirichlet Process(DP). The second level is a multinomial distribution, which is represented as G_0, and G_0 is shared by all the documents in the data collection. Specifically,G_0 can be deemed as the aspect distributions we want to generate from the review collection. The first level is a series of multinomial distributions θ_d, which control word-aspect assignments within each document. θ_d are private to each of the documents, and generated by G_0. If we explain HDP from the Chinese Restaurant Process(CRP)[23] perspective, each document can be treated as a restaurant, and words in the document are the customers, who will sit tables within this restaurant. The first customer will sit the first table, and the n^{th} customer will sit at table t with probability of $\frac{c_t}{n-1+\alpha}$, where c_t is the number of words sat in table t, and create a new table with probability of $\frac{\alpha}{n-1+\alpha}$. Similarly, tables will order dishes. A table will order a dish k with probability of $\frac{s_k}{m-1+\gamma}$, where s_k is the number of tables ordered dish k, and create a new dish with probability of $\frac{\gamma}{m-1+\gamma}$. α and γ are the hyper-parameters in the model. The table and dish here can be treated as the local and global aspect distributions. The graphic model of HDP is shown in Figure 1(a).

Just as shown in the model, the probability of a customer sitting at a table is only proportional to the number of other customers already in that table[24]. Such assignments are kind of random and ignore the context information. Aiming to solve this problem, Blei has proposed a Distance Dependency Chinese Restaurant Process(Dist-CRP) [24], which takes the distance information into consideration in image process. Although it improves the performance by clustering close pixels within one document, Dist-CRP ignores the words' global co-occurrence information. Actually, one reason why LDA can achieve a satisfying result in topic detection is that it exploits the words' co-occurrence information well, while HDP, including Dist-CRP, has over-

looked such information by implementing the private $\boldsymbol{\theta_d}s$. Thus, this paper has proposed a Semantic Dependency Dirichlet Process (SDDP) for sentiment analysis. SDDP considers not only the local distance information, but also the global co-occurrence information. Based on SDDP, we construct two kinds of models, one word model (W-SDDP) and one phrase model (P-SDDP). The experiment results show that SDDP based models perform better than LDA, HDP and their extended models.

3 Model Description.

3.1 Similarity Dependency Dirichlet Process.

Words' co-occurrence and distance are the two perspectives reflecting two words' semantic similarity. Thus, this paper uses formula(3.1) as the function to calculate two words' semantic similarity.

(3.1)
$$sim(w_i, w_j) = m * \sum_{d=1}^{D} \sum_{i,j=0}^{M_d} (\frac{e^{-|i-j|} - e^{-M_d}}{e^{-1} - e^{-M_d}} * \frac{1}{c(w_i) + c(w_j)})$$

where D is the number of Document, M_d is the length of document d, w_i represents the word appearing in document d's i^{th} position, $c(w_i)$ denotes the total number of times w_i occurring in the document collection, and m is the normalization coefficient to ensure $sim(w_i, w_j)$ ranges in [0,1]. Thus, the more often w_i and w_j appear in the same document, or the closer w_i and w_j present, the larger $sim(w_i, w_j)$ is.

SDDP utilize $sim(w_i, w_j)$ to assign words. Given $w_1, w_2, .w_{n-1}$ within a document, the table assignment a_n of the n^{th} word/phrase follows the formula (3.2), where t represents the table assignments, and $count(t)$ means the number of words that have been assigned to table t.

$$p(a_n = t|\boldsymbol{a_{1:n-1}}, \boldsymbol{w_{i\in t}}, \alpha, sim(\cdot)) \propto \begin{cases} \frac{\sum_{i\in t} sim(w_i, w_n)}{count(t)} \text{(if t exists)} \\ \alpha \quad \text{(if t is new)} \end{cases} \tag{3.2}$$

Similarly, in the second level, given the all the tables $t_1, t_2, .t_{m-1}$ generated among the whole document collection, the topic assignment z_m for the m^{th} table can be represented as formula(3.3), where k represents the topic assignment, and $count(k)$ is the number of tables assigned to this topic.

$$p(z_m = k|\boldsymbol{z_{1:m-1}}, \boldsymbol{t_{j\in k}}, \gamma, sim(\cdot)) \propto \begin{cases} \frac{\sum_{j\in k} sim(t_j, t_m)}{count(k)} \text{(if k exists)} \\ \gamma \quad \text{(if k is new)} \end{cases} \tag{3.3}$$

3.2 Two Derivation Models.

Just as mentioned in related work part, there are mainly two types of models dealing with sentiment analysis, namely the word model and the phrase model, and in addition, the word model can be classified into the pure word model and the mixture word model. Considering the mixture word model is a kind of combination of the phrase model and the pure word model, and in order to make a complete contrast, this paper extends SDDP into a pure word model (W-SDDP) and a phrase model (P-SDDP).

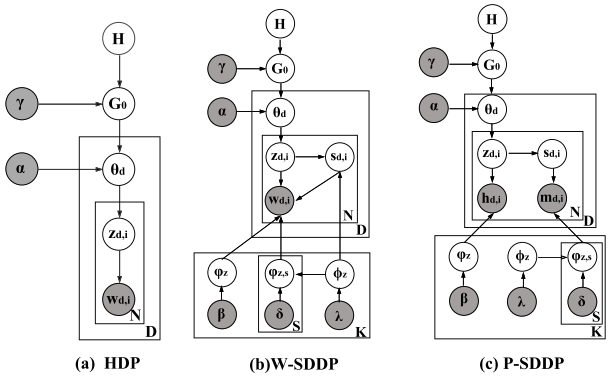


Figure 1: Graphic Models

W-SDDP assumes each document is composed by a series of independent words, and each word conveys both aspect and sentiment information. While, P-SDDP assumes each document is composed by a series of independent phrase, which can be represented as $< head\ word, modifier\ word >$, and the head word conveys the aspect information, and the modifier word conveys the sentiment information. Figure 1(b) and Figure 1(c) show the graphic models for W-SDDP and P-SDDP. Table 1 shows the explanations for the annotation in Figure 1.

In generative process, W-SDDP and P-SDDP have a lot in common. To be concise, we use the same flow to introduce the generative process, but highlight the different parts for these two models. Their generative process can be shown as follows:

Step 1: Define a baseline H for global aspect generation. Here we choose a uniform distribution as H. Draw a distribution G_0 from H according to SDDP parametrized by γ. $G_0 \sim SDDP(H, \gamma)$.

Step 2: For each aspect, draw a word-aspect distribution φ_k according to a Dirichlet distribution parametrized by β. $\varphi_k \sim Dir(\beta)$.

Step 3: For each aspect, draw sentiment distributions $\varphi_{k,s}$ according to a Dirichlet distribution parametrized by δ_s. $\varphi_{k,s} \sim Dir(\delta_s)$.

Step 4: For each document d:

(4.1) Draw a multinomial distribution θ_d from G_0 according to SDDP parametrized by α. $\theta_d \sim SDDP(G_0, \alpha)$.

(4.2) For the i^{th} word or phrase in document d:

(4.2.1) Draw an aspect assignment $z_{d,i}$ according to θ_d.

(4.2.2) Draw a sentiment distribution ϕ_z according to a Dirichlet distribution parametrized by λ. $\phi_z \sim Dir(\lambda)$.

(4.2.3) Draw a sentiment assignment $s_{d,i}$ according to ϕ_z.

Table 1: Annotation of the Graphic Model

D	The number of documents
N	The number of words/phrases
K	The number of aspects
S	The number of sentiments
H	The baseline distribution to generate G_0
G_0	The global aspect distribution shared by all the documents
θ_d	The The local aspect distribution of document d
φ_z	The word distribution within an aspect z
$\varphi_{z,s}$	The word distribution of a sentiment s in aspect z
ϕ_z	The sentiment distribution within an aspect z
$z_{d,i}$	The aspect assignment of the i^{th} word/phrase in the d^{th} document
$s_{d,i}$	The sentiment assignment of the i^{th} word/phrase in the d^{th} document
$w_{d,i}$	The i^{th} word in the d^{th} document
$h_{d,i}$	The head word of the i^{th} phrase in the d^{th} document
$m_{d,i}$	The modifier word of the i^{th} phrase in the d^{th} document
α	The hyper parameter for local aspect assignment
β	The hyper parameter for word allocation within an aspect
γ	The hyper parameter for global aspect assignment
λ	The hyper parameter for sentiment distribution within an aspect
δ	The hyper parameter for word distribution within a sentiment

For W-SDDP:

(4.2.4) Generate a word $w_{d,i}$ according to φ_z and $\varphi_{z,s}$. $w_{d,i} \sim \varphi_z, \varphi_{z,s}$.

For P-SDDP:

(4.2.4) Generate the head of $p_{d,i}$ according to φ_z. $h_{d,i} \sim \varphi_z$.

(4.2.5) Generate the modifier of $p_{d,i}$ according to $\varphi_{z,s}$. $m_{d,i} \sim \varphi_{z,s}$.

3.3 Model Inference. We use Gibbs Sampling to infer both of W-SDDP and P-SDDP. Considering W-SDDP and P-SDDP share very similar sampling process, and the only difference is that W-SDDP use words to sample both aspects and sentiments, while P-SDDP use head words to infer aspects and modifier words to infer sentiments. To make the paper concise, we use W-SDDP as the example to show how to make the inference, as for P-SDDP, please replace $w_{d,i}$ with $h_{d,i}$ in aspect inference, and replace $w_{d,i}$ with $m_{d,i}$ in sentiment inference.

At the first level, each word needs to be assigned to different tables. This assignment can be realized by the formula(3.4), where $a_{d,i}$ denotes the table assignment of $w_{d,i}$, $\boldsymbol{a^{-d,i}}$ represents other words' table assignments in document d except $w_{d,i}$, k represents the aspects' word distributions, $ave(sim(w_{d,i}, w_n))$ denotes to $\frac{\sum_{i \in t} sim(w_n, w_i)}{count(t)}$in formula(3.2), and $g_k^{-w_{d,i}}(k_t)$ denotes to the word distribution in the aspect which table t has been assigned to.

$$p(a_{d,i} = t|\boldsymbol{a^{-d,i}}, \boldsymbol{k}) = \tag{3.4}$$
$$(\sum_{t=1}^{T} \frac{\sum_{n \in t} ave(sim(w_{d,i}, w_n))}{\sum_{t=1}^{T}\sum_{n \in t} ave(sim(w_{d,i}, w_n)) + \alpha} * g_k^{-w_{d,i}}(k_t))$$
$$+ \frac{\alpha}{\sum_{t=1}^{T}\sum_{n \in t} ave(sim(w_{d,i}, w_n)) + \alpha}$$

Similarly, table will be assigned to different aspects in the second level, the inference process can be shown as formula (3.5), where $z_{d,t}$ denotes to the aspect assignment for the t^{th} table in document d, $ave(sim(t_{d,j}, t_m))$ denotes to $\frac{\sum_{m \in k} sim(t_m, t_j)}{count(k)}$ in formula (3.3), and $g_k^{-t_{d,j}}(k)$ denotes to the word distribution in aspect k.

$$p(z_{d,j} = k|\boldsymbol{z^{-d,j}}, \boldsymbol{k}) = \tag{3.5}$$
$$(\sum_{k=1}^{K} \frac{\sum_{m \in k} ave(sim(t_{d,j}, t_m))}{\sum_{k=1}^{K}\sum_{m \in k} ave(sim(t_{d,j}, t_m)) + \gamma} * g_k^{-t_{d,j}}(k))$$
$$+ \frac{\gamma}{\sum_{k=1}^{K}\sum_{t \in k} ave(sim(t_{d,j}, t_m)) + \gamma}$$

$g_k^{-w_{d,i}}(k)$can be inferred as formula(3.6) , where $N_{k,w}^{-w_{d,i}}$ is the word count of w in aspect k except $w_{d,i}, N_k^{-w_{d,i}}$ is the number of words in aspect k except $w_{d,i}$ and V is the length of word vocabulary.

$$g_k^{-w_{d,i}}(k) = \frac{N_{k,w}^{-w_{d,i}} + \beta}{N_k^{-w_{d,i}} + V * \beta} \tag{3.6}$$

After aspect inference, a sentiment needs to be chosen for the very word or phrase under this aspect. We apply a Dirichlet allocation as the prior for sentiment distribution. Under the aspect k, the sentiment inference can be made as formula (3.7), where $N_{k,s,w}^{-w_{d,i}}$is the number of word w has been assigned to sentiment s under aspect k except $w_{d,i}$, $N_{k,s}^{-w_{d,i}}$ the number of words have been assigned to sentiment s under aspect k except $w_{d,i}$, $N_k^{-w_{d,i}}$ is the word count of aspect k except $w_{d,i}$, and S is the number of sentiment.

$$p(s_{d,i} = s|k) = \frac{N_{k,s,w}^{-w_{d,i}} + \delta_s}{N_{k,s}^{-w_{d,i}} + V * S} * \frac{N_{k,s}^{-w_{d,i}} + \lambda}{N_k^{-w_{d,i}} + S * \lambda} \tag{3.7}$$

So, in summary the inference process can be represented as the formula (3.8).

$$p(z_{d,i} = k, s_{d,i} = s|\boldsymbol{parameters}) = \tag{3.8}$$
$$((\sum_{t=1}^{T} \frac{\sum_{n \in t} ave(sim(w_{d,i}, w_n))}{\sum_{t=1}^{T}\sum_{n \in t} ave(sim(w_{d,j}, w_n)) + \alpha} * g_k^{-w_{d,i}}(k_t))$$
$$+ \frac{\alpha}{\sum_{t=1}^{T}\sum_{n \in t} ave(sim(w_{d,i}, w_n)) + \alpha}$$
$$* ((\sum_{k=1}^{K} \frac{\sum_{m \in k} ave(sim(t_{d,i}, t_m))}{\sum_{k=1}^{K}\sum_{m \in k} ave(sim(t_{d,i}, t_m)) + \gamma} * g_k^{-t_{d,i}}(k))$$
$$+ \frac{\gamma}{\sum_{k=1}^{K}\sum_{m \in k} ave(sim(t_{d,i}, t_m)) + \gamma} * G_0))$$
$$* (\frac{N_{k,s,w}^{-w_{d,i}} + \delta_s}{N_{k,s}^{-w_{d,i}} + V * S} * \frac{N_{k,s}^{-w_{d,i}} + \lambda}{N_k^{-w_{d,i}} + S * \lambda})$$

4 Experiment and Evaluation.

4.1 Dataset and Model Description. All the experiments and evaluations are tested on one or some of the six datasets shown in Table 2. We used different datasets for different experiment or evaluation purposes. The models to be tested are LDA[7], HDP[17], JST[8], ASUM[11], MaxEnt-LDA[10], JAS[22], and our two models: W-SDDP and P-SDDP.

Table 2: Dataset List

NO.	Dataset Content	Source	Volume	Labelled
1[25]	Restaurant	Citysearch	3400 sentences	Yes
2[21]	Coffee Machine	Amazon	3000 reviews	No
3[21]	Laptop	Amazon	3000 reviews	No
4[26]	Car	tripAdviser	3000 reviews	No
5[26]	Hotel	tripAdviser	3000 reviews	No
6	Restaurant	Yelp.com	350 reviews	No

4.2 Experiment Settings. P-SDDP functions only when the documents can be decomposed into a series of phrases, which are presented as $< head\ word, modifier\ word >$. We use Stanford Dependency Parser (SDParser)[27] to process the datasets for phrase model. Given a sentence, SDParser can find the word pairs that have modification relations. According to the results provided by SDParser, the overall relationships and patterns we use are listed as follows, where A in $< A, B >$ denotes the head word and B in $< A, B >$ denotes the modifier word. For detailed information, please refer to SDParser manual book[28].

$Adjectival Modifier : amod(A, B) \rightarrow < A, B >$

$Adjectival\ \ Complement : acomp(A, B) + nsubj(A, C) \rightarrow < C, B >$

$Copula : cop(A, B) + nsubj(A, C) \rightarrow < C, A >$

$Direct\ \ Object : dobj(A, B) + nsubj(A, C) \rightarrow < B, A >$

$And :< A, B > +conj\_and(A, C) \rightarrow < C, B > or < A, B > +conj\_and(B, C) \rightarrow < A, C >$

$Negation\ \ Modifier :< A, B > +neg(B, not) \rightarrow < A, not + B >$

$Noun\ \ Compound :< A, B > +nn(A, C) \rightarrow < C + A, B >, or < A, B > +nn < C, A \rightarrow < A + C, B >$

$Agent\ \ Relationship : agent(A, B) \rightarrow < B, A >$

$Nominal\ \ Subject : nsubj(A, B) \rightarrow < B, A >$

$Infinitival\ \ Modifier : infmod(A, B) \rightarrow < A, B >$

$Passive\ \ Nominal\ \ Subject : nsubjpass < A, B > \rightarrow < B, A >$

$Participial\ \ Modifier : partmod(A, B) \rightarrow < A, B >$

$Controlling\ \ Subject : xsubj(A, B) \rightarrow < B, A >$

4.3 Prior Knowledge. We use MQPA as the sentiment dictionary to facilitate the hyper parameter settings for sentiment distribution. MQPA has provided the polarities for each word, but not the explicit score. Thus, we make the following rules:

If a word is tagged as "positive" and "strongsubj", $\delta_{positive} = 0.8, \delta_{negative} = 0.1, and, \delta_{neutral} = 0.1$

If a word is tagged as "positive" and "weaksubj", $\delta_{positive} = 0.6, \delta_{negative} = 0.1, and, \delta_{neutral} = 0.3$

If a word is tagged as "negative" and "strongsubj", $\delta_{positive} = 0.1, \delta_{negative} = 0.8, and, \delta_{neutral} = 0.1$

If a word is tagged as "negative" and "weaksubj", $\delta_{positive} = 0.1, \delta_{negative} = 0.6, and, \delta_{neutral} = 0.3$

If a word is tagged as "neutral" and "strongsubj", $\delta_{positive} = 0.1, \delta_{negative} = 0.1, and, \delta_{neutral} = 0.8$

If a word is tagged as "neutral" and "weaksubj", $\lambda_{positive} = 0.6, \lambda_{negative} = 0.2, and, \lambda_{neutral} = 0.2$

For other hyper parameters in the models, we employ the standard and out of box settings without any tuning to our datasets. For the six comparison models, all the hyper parameters are set as default values, and for W-SDDP, and P-SDDP, α and γ are set with the same magnitude of the similarity value, and β=λ=0.05.

4.4 Evaluation

Evaluation with Golden Standard. First, we use golden standard to evaluate our models. The first dataset in Table 2 is the one with golden standard. All the words in this dataset have been manually annotated to six aspects, namely Food, Staff, Price, Ambience, Anecdote, and Miscellaneous, and three sentiments: Positive, Negative and Neutral.

Models like JST, ASUM, and JAS, mix all the aspect words and sentiment words together, so we extract the noun words as the aspect words and others as sentiment words to map with the golden standard. Models like MaxEnt, LDA and HDP, do not provide sentiment polarities, so we can only compare them on the general sentiment level without distinguishing the specific sentiment types.

We use precision to measure to what degree each model can correctly select the aspect words and the corresponding sentiment words. The results are shown in Table 3, Table 4 and Table 5. The blanks in the tables mean that we could not find the related aspect or sentiment from the model's results.

Table 3: Aspect Comparison among the Popular Models

	LDA	HDP	ASUM	JST	Max-Ent	JAS	W-SDDP	P-SDDP
Food	0.639	0.806	0.751	0.632	0.808	0.779	0.760	**0.817**
Staff	0.429	0.460	0.411	0.299	0.559	0.527	0.563	**0.655**
Price	–	0.353	0.278	–	0.232	0.351	0.366	**0.494**
Ambience	0.412	0.452	0.347	0.226	0.299	0.451	0.469	**0.545**
Anecdote	0.379	0.444	0.259	0.188	0.397	0.443	0.450	**0.450**
Miscellaneous	0.441	0.471	0.504	0.347	0.330	0.532	0.565	**0.590**

From Table 3 we can find that P-SDDP performs the best in identifying all the six aspects, and W-SDDP also outperforms other non-SDDP based models except

in detecting "Food", which is just slightly lower.

Table 4: Sentiment Comparison among the Models with Sentiment Polarity

		ASUM	JST	JAS	W-SDDP	P-SDDP
Food	+	0.655	0.461	0.658	**0.822**	0.786
	-	0.368	0.225	0.224	**0.440**	0.400
	*	0.104	0.064	–	0.136	**0.304**
Staff	+	0.445	0.241	0.243	**0.667**	0.662
	-	0.388	0.164	0.322	0.438	**0.651**
	*	0.022	0.037	–	0.071	0.063
Price	+	–	–	0.255	0.333	**0.431**
	-	0.150	–	0.088	**0.333**	0.273
	*	–	–	–	0.000	0.000
Ambience	+	–	–	0.273	**0.701**	0.565
	-	0.174	–	0.124	0.286	**0.400**
	*	0.056	0.029	–	0.078	**0.158**
Anecdote	+	–	0.089	0.093	**0.500**	0.256
	-	–	–	0.143	**0.333**	0.250
	*	0.243	0.113	–	0.200	**0.444**
Miscellaneous	+	0.302	0.241	0.227	**0.636**	0.583
	-	0.218	–	0.176	0.250	**0.400**
	*	0.219	–	–	**0.500**	0.231

Table 5: Sentiment Comparison among Models without Sentiment Polarity

	LDA	HDP	MaxEnt	W-SDDP	P-SDDP
Food	0.230	0.161	0.221	**0.602**	0.530
Staff	0.197	0.090	0.205	**0.583**	0.391
Price	–	0.059	0.134	**0.301**	0.263
Ambience	0.187	0.082	0.107	**0.440**	0.406
Anecdote	0.164	0.083	0.131	0.281	**0.333**
Miscellaneous	0.190	0.000	0.091	0.452	**0.500**

From Table 4 and 5, we can find the both W-SDDP and P-SDDP outperform other models, and they beat each other alternatively.

In this section, we find that both of W-SDDP and P-SDDP perform better than other models when evaluating via datasets with golden standard.

Evaluation with Perplexity. Although we can evaluate models via golden standard, the quantity of labelled datasets is very limited. It is not persuasive by just testing the results on only one dataset. Another way to evaluate the performances of the models is to apply *Perplexity* on unlabelled datasets.

We test the 8 models on 4 datasets, from the 2^{nd} to 5^{th} dataset in Table 2. For each model on each dataset, we change the initial number of aspects from 10 to 100 in intervals of 10, and choose the lowest perplexity as the value to compare. The results are shown in Figure 2. From Figure 2, we can see that P-SDDP always has the lowest perplexity on all the datasets, followed by W-SDDP. Perplexity based evaluation indicates that

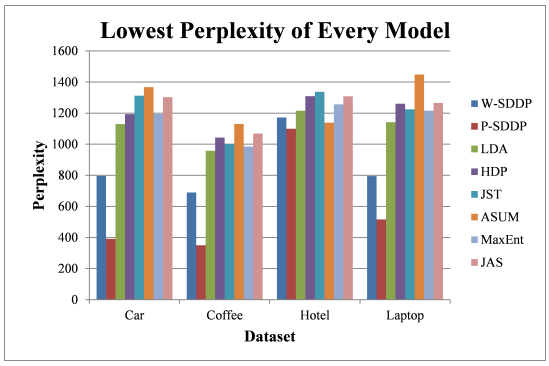


Figure 2: Perplexity Comparison among Models

W-SDDP and P-SDDP have better performances in inferring the words in the datasets. Comparing to W-SDDP, P-SDDP has an even better performance.

5 Applications in Recommendation System.

In this section, we will show the practical applications of W-SDDP and P-SDDP in Recommendation System.

5.1 Aspect Specific Dictionary Construction. Both W-SDDP and P-SDDP can help to construct the aspect- specific dictionary. Table 7 and Table 8 show the major aspects and their corresponding sentiment words detected by W-SDDP and P-SDDP respectively from the first dataset in Table 2. From Table 7 and Table 8, we find for a restaurant, the commonly mentioned aspects are Food (including Chinese food, Japanese food etc.), Atmosphere, Service and Staff. In addition, for each aspect, people might use different sentiment words to express their feelings. The sentiment word like "Oily" conveys a negative sentiment for Chinese food, but "watery" conveys a positive sentiment. Similarly, the word "Repeatable" conveys a negative sentiment when describes the staff, but it may convey a positive sentiment in other scenario.This result can be implemented in Review Recommendation System to help detect the good reviews to recommend to customers.

5.2 Online Review Summarization. In many cases, we want to be recommend a product by a certain aspect. Taking restaurant as an example, by explicit providing the aspect and sentiment summarization information can help to recommend a restaurant more precisely.

We implement both W-SDDP and P-SDDP on the

Table 6: Result of W-SDDP

Aspect	Sentiment	
Atmosphere-Service: Service,Place,Time, Menu,Atmosphere, Staff,Dishes,Drinks	+	Nice,Great,Wonderful,Decent, Popular,Relax,Superb,Friendly
	-	Dim,Horrible,Mediocre Disappointing,Crowded,Poorly Slow,Worst
Food-Pizza: Pizza,Crust,Slice, Cheese,Williamsburg, Mushroom	+	Adorable,Delicate,Crisp,Fancy Best,Pretty,Supreme,Perfect
	-	Horrific,Vomit,Disgusting Complaints,Tiny,Gross, Expensive, Not-Special
Food-Japan& China: Sushi,Sichuan,Roll Eel, Sea, Chongqing, Fish, Chinatown Shanghai	+	Heavenly,Rejoice,Special,Best, Amazingly,Favorite,Fresh, Elegant
	-	Mock, Rigid, Dull, Overdone, Fatty, Weird, Poor, Not-Fresh
Food-USA: Bagel, Bagels, Coffee, Freeze, Cream Cheeses,Takeaway Mayo	+	Colossal, Outstanding, Best, Plentiful, Big, Original, Pleasantly, Fabulous
	-	Strange, Pricey, Not-Nice, Not-Authentic, Bland, Spot, Disappointed
Staff: Table, Dinner, Waitstaff, Minute, Service, Minutes, Bartender, Waiter	+	Hospitable, Experienced, Nice, Stylish, Not-Unable, Helpful, Ready, Attentive
	-	Confused, Not-Amazed, Annoying, Not-Competent, Unpleasant, Noisy, Clumsy, Pretentious

Table 7: Result of P-SDDP

Aspect	Sentiment	
Atmosphere-Service: Service, Place, Dishes, Atmosphere, Night, Staff	+	Reasonable, Accommodating, Friendly, Relaxing, Romantic, Excellent, Expected, Cool
	-	Rude, Noisy, Disappointing, Biting, Dark, Poor, Drafty, Slow
Food-Pizza: Pizza,Slice,Crust, Ingredients,Codfish Addition,Lobster,Pie	+	Crisp, Fresh, Thin, Expanded, Fresh-Tasting,Well-Seasoned, Delicious,Tasty
	-	Shredded, Vomit-Inducting, Not-Topped,Skimp,Not-Want, Common,Bitter,Bland
Food-Japan: Sushi,Rice,Tuna, Fish, Sauces, Scallop, Roll,Appetizer	+	Spicy,Matches,Please, Healthy-Looking, Recommended, Favorite Refreshing, Superb
	-	Disgusting,Flavorless,Not-Exciting,Broken,Horrid, Rough,Murky,Awful
Food-China: Pork,Soup, Dumpling,Chicken, Shanghai, Shanghainese, Scallion, Eggplant	+	Tasting,Traditional,Amazing, Watery,Love,Wonderful, Authentic,Complimentary
	-	Sour, Mock, Lacking, Horrible, Overcompensate, Oily, Overpriced,Small
Staff: Staff, Service, Manager,People, Cooks,Menu,Tables, Reservation	+	Friendly, Great, Enthusiastic, Attentive, Helpful, Knowledgeable, Wonderful
	-	Not-recommend,Lies,Bad, Unavailable, Repeatable, Unpleasant, Not-inspired, Lazy

6^{th} dataset, and find there are 5 aspects people mention a lot toward this restaurant, namely Chicken & Waffles, the signature dish of this restaurant, Food rather than Chicken & Waffles, Atmosphere, Service and Price. For most aspects, like food (including Chicken & Waffles), atmosphere, and service, people tend to give a positive judgement. While for the price, the negative sentiment proportion is a little larger. Thus, to a consumer who emphasize the food or service quality, we can recommend this restaurant, but to a consumer who cares about the price, we may ignore this restaurant.

6 Discussion and Conclusion.

6.1 Comparison between W-SDDP and P-SDDP. Section 4.4 has proved that W-SDDP and P-SDDP indeed outperform other models. In this part, we will compare W-SDDP and P-SDDP to see which one is better in application. All the experiments in this part are conducted on the first dataset in Table 2.

Table 6 shows that comparing to W-SDDP, P-SDDP has a lower converged aspect number and a lower perplexity. In this sense, P-SDDP performs better than W-SDDP.However, P-SDDP loses a lot of information. In Table 6, we can see that P-SDDP loses near to 32.5% word token in phrase process, because some words could not be paired up to phrases, and are removed by the parser.

Thus, in real use, one needs to balance these two models. P-SDDP will give a more concise and better result, but lose considerable amount of information. W-SDDP keeps all the information, but might bring some noise to the results.

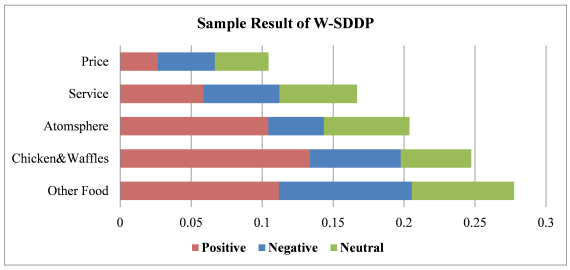
Table 8: Comparison between W-SDDP and P-SDDP

	W-SDDP	P-SDDP
Number Of Tokens	30035	20274
Converged Aspect Number	20-30	8-10
Perplexity	Around 900	Around 300

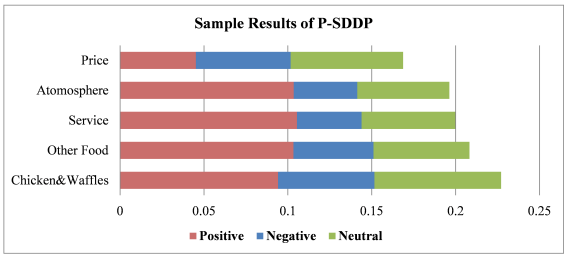
6.2 Conclusion. *Sentiment Analysis* is a core component for review recommendation system. This paper has constructed a Similarity Dependency Dirichlet Process (SDDP) as a novel model for sentiment analysis. SDDP has solved the aspect number specification problem encountered in LDA, and improves the aspect/sentiment detection performance by replacing the random word assignment mechanism with similarity based word assignment.Based on SDDP, two models are built. One is a word model W-SDDP, and a phrase model P-SDDP. Evaluation results show that both W-SDDP and P-SDDP perform well on various datasets.

References

[1] Ghose, A, P. G. Ipeirotis,*Designing novel review ranking systems: predicting the usefulness and impact of reviews*,



(a) Sample Result of W-SDDP



(b) Sample Result of P-SDDP

Figure 3: Sample Results

Proceedings of the ninth international conference on Electronic commerce, ACM, 2007.

[2] Momeni, E., K. Tao, B. Haslhofer,G.-J. Houben, *Identification of Useful User Comments in Social Media: A Case Study on Flickr Commons*,Joint Confernece on Digital Library, 2013.

[3] S. R. Das and M. Y. Chen,*Yahoo! for Amazon: Sentiment extraction from small talk on the web*, Management Science, vol. 53, pp. 1375-1388, 2007.

[4] L. Dini and G. Mazzini,*Opinion classification through information extraction*, in Proceedings of the Conference on Data Mining Methods and Databases for Engineering, Finance and Other Fields (Data Mining), 2002, pp. 299-310.

[5] T. Nasukawa and J. Yi,*Sentiment analysis: Capturing favorability using natural language processing*, in Proceedings of the 2nd international conference on Knowledge capture, 2003, pp. 70-77.

[6] B. Pang, L. Lee, and S. Vaithyanathan,*Thumbs up?: sentiment classification using machine learning techniques*, in Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, 2002, pp. 79-86.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan,*Latent Dirichlet allocation*, Journal of Machine Learning Research, vol. 3, pp. 993-1022, May 15 2003.

[8] C. Lin and Y. He,*Joint sentiment/topic model for sentiment analysis*, in Proceedings of the 18th ACM conference on Information and knowledge management, 2009, pp. 375-384.

[9] I. Titov and R. McDonald,*Modeling online reviews with multi-grain topic models*, in Proceedings of the 17th international conference on World Wide Web, 2008, pp. 111-120.

[10] W. X. Zhao, J. Jiang, H. Yan, and X. Li,*Jointly modeling aspects and opinions with a MaxEnt-LDA hybrid*, in Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, 2010, pp. 56-65.

[11] S. Brody and N. Elhadad,*An unsupervised aspect sentiment model for online reviews*, in Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 2010, pp. 804-812.

[12] T. J. Zhan and C. H. Li,*Semantic Dependent Word Pairs Generative Model for Fine-Grained Product Feature Mining*, PAKDD 2011, Shenzhen China, May 24-27,2011, pp. 460-475, 2011.

[13] W. Jin, H. H. Ho, and R. K. Srihari,*OpinionMiner: a novel machine learning system for web opinion mining and extraction*, in Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 2009, pp. 1195-1204.

[14] H. Guo, H. Zhu, Z. Guo, and Z. Su,*Domain customization for aspect-oriented opinion analysis with multi-level latent sentiment clues*, in Proceedings of the 20th ACM international conference on Information and knowledge management, 2011, pp. 2493-2496.

[15] J. Si, A. Mukherjee, B. Liu, Q. Li, H. Li, and X. Deng,*Exploiting Topic based Twitter Sentiment for Stock Prediction*, in ACL, 2013, pp. 24-29.

[16] S. Kim, J. Zhang, Z. Chen, A. H. Oh, and S. Liu,*A Hierarchical Aspect-Sentiment Model for Online Reviews*, in AAAI, 2013.

[17] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei,*Hierarchical Dirichlet Processes*,Journal of the American Statistical Association, vol. 101, pp. 1566-1581, 2006.

[18] D. M. Blei and P. I. Frazier,*Distance dependent Chinese restaurant processes*, The Journal of Machine Learning Research, vol. 12, pp. 2461-2488, 2011.

[19] S. Moghaddam and M. Ester,*On the design of LDA models for aspect-based opinion mining*, in Proceedings of the 21st ACM international conference on Information and knowledge management, 2012, pp. 803-812.

[20] S. Moghaddam and M. Ester,*ILDA: Interdependent LDA Model for Learning Latent Aspects and Their Ratings From Online Product Reviews*, SIGIR'11, pp. 665-674, 2011.

[21] Y. Jo and A. Oh,*Aspect and Sentiment Unification Model for Online Review Analysis*, WSDM'11 February 9-12, Hong Kong, China, 2011.

[22] X. Xu, S. Tan, Y. Liu, X. Cheng, and Z. Lin,*Towards Jointly Extracting Aspects and Aspect-Specific Sentiment Knowledge*, CIKM'12, October 29-November , 2012 Maui, HI, USA, 2012.

[23] D. Blei,*COS 597C: Bayesian nonparametrics*,Lecture Notes in Priceton University. http:// www.cs.princeton.edu/ courses/ archive/ fall07/ cos597C/ scribe/ 20070921.pdf, 2007.

[24] D. M. Blei and P. Frazier*Distance Dependent Chinese Restaurant Process*, Proceedings of the 27th International Conference on Machine Learning, Haifa, Isreal, 2010.

[25] G. Ganu, N. Elhadad, and A. Marian,*Beyond the Stars: Improving Rating Predictions using Revew Text Cotent*, Twelfth International Workshop on the Web and Databases, Providence, Rhode, Island, USA, 2009.

[26] K. Ganesan and C. Zhai,*Opinion-based entity ranking*, Information retrieval, vol. 15, pp. 116-150, 2012.

[27] M.C. De Marneffe, B. MacCartney, and C. D. Manning,*Generating typed dependency parses from phrase structure parses*, in Proceedings of LREC, 2006, pp. 449-454.

[28] M.C. De Marneffe and C. D. Manning,*Stanford typed dependencies manual*, URL http://nlp. stanford. edu/software/dependencies manual. pdf, 2008.

Recommendations on a Knowledge Graph

László Grad-Gyenge*
laszlo.grad-gyenge@tuwien.ac.at

Peter Filzmoser†
peter.filzmoser@tuwien.ac.at

Hannes Werthner‡
hannes.werthner@tuwien.ac.at

Abstract

Most recommender system methods derive the user preferences from predefined information sources. For example, collaborative filtering is based on user rating values on items. Predefining information sources constrains the quality of the recommendation result by restricting the amount of information the recommendation method can operate on. In this paper we introduce an adaptive rating estimation method, which is capable to incorporate heterogeneous information sources and improves the recommendation quality.

To represent heterogeneous information, a graph based knowledge base is introduced. Recommendations are calculated with our novel method, recommendation spreading. Comparing recommendation spreading to collaborative filtering on the MovieLens 1M dataset shows that our method is able to combine heterogeneous information sources to provide higher coverage and the same rating estimation error. Furthermore, recommendation spreading is a potential method to overcome the cold start problem.

1 Introduction.

To enhance recommendation quality, several information sources have been involved into the recommendation process by the recommender systems community. Most of the methods we have found during our research explicitly define the type of information sources to calculate the recommendations from. To mention the most popular ones, recommendations can be derived from user preferences on items, user attributes, product attributes, social network, product description, user interaction, ontology information, purchase history or expert knowledge. One of our goals is to develop a recommendation framework providing an information representation method which is general enough to represent and to integrate information from various types of information sources. Our intention is to provide an information representation method which can act as a stable basis for the elaboration of more general recommender systems. By generality we mean methods, which are capable to weigh appropriately the available information source types instead of working with predefined ones.

The Machine Learning community has evolved several adaptive classification, clustering and prediction techniques. By adaptiveness we mean that after a certain period of training the method learns the underlying structure or the features of the data and makes its predictions and decisions based on the previously learnt model. By defining the knowledge base to contain a variety of information sources, our intention is to introduce recommendation methods which are influenced by machine learning in the sense they are adaptive and can be trained on past data. We think that this area is a promising direction of research. In this paper we present our first, spreading activation based calculation method, which is a promising step in this direction. Spreading activation is a rarely used technique in this field. For example Hussein et al. utilize activation spreading to deliver explanations to the user [10].

In this paper we will introduce our heterogeneous knowledge base which is capable to incorporate several information source types. The knowledge base is introduced as a directed, labeled, restricted hypergraph. We also define a recommendation calculation method which provides higher quality recommendations than collaborative filtering does. This way we prove on a working example that the involvement of an increased and heterogeneous amount of information sources can lead to higher quality recommendations.

Section 2 contains an overview of recommendation techniques, which operate with graph related representation methods. In Section 3 we provide a formal definition of our knowledge base. Section 4 describes our calculation method, which is based on spreading activation. In Section 5 we describe how we evaluated our pivot method and present our evaluation results. Section 6 concludes the paper and gives an insight into our plans for the future.

*Electronic Commerce Group, Vienna University of Technology

†Department of Statistics and Probability Theory, Vienna University of Technology

‡Electronic Commerce Group, Vienna University of Technology

2 Graph Based Representation.

A trend is visible to graph based representation in recommender systems. Collaborative filtering is the best-known recommendation method. In most cases the knowledge base of collaborative filtering is modeled with a matrix [16]. Collaborative filtering estimates user preferences on items based on the already known preference values with rather good results. Such preferences can be explicit (user rating, user like, purchase) or implicit (commenting an item, viewing item details, mentioning an item in a post). A visible trend in the field of recommender systems is to involve additional information sources to for example collaborative filtering in order to improve its performance. To represent this information, various methods were developed.

Konstas et al. [13] recommend music tracks for users. To improve the recommendation quality, in addition to item rating they involve user issued tags and social relationships between users into the knowledge base. They represent the different kind of relations as `UU` (user-user), `UTr` (user-track), `UTg` (user-tag) and `TrTg` (track-tag) with a partitioned matrix. Regarding information type content, each of those partitions of the matrix are homogeneous but the overall matrix is heterogeneous. Hidasi et al. [9] take context information into the recommendation process. To represent the information they utilize a tensor algebra, which can be treated as the generalization of matrices into higher dimension. We would like to mention here that in both of the above mentioned cases the matrix representation can be substituted with an equivalent, graph based representation.

Kazienko et al. [12] use a multi-layered graph to represent the information necessary to compute recommendations. To estimate user preferences, their method relies on contact lists, tags, groups, favorites, opinions and social networks. They represent each information type on a separate layer, where each layer contains a graph representing homogeneous information. The layered approach could be mapped to a unified but heterogeneous graph by adding a type attribute to the respective graph nodes and edges.

Furthermore, involving trust networks into the recommendation process was a visible trend of the recommender community in the years 2004-2006 [5][17][14][11]. A straightforward representation of a trust network is a directed graph. A directed graph would encode the users as nodes and the trust relationships as edges. Because of its explicitness, the involvement of this kind of information source has a good chance to increase recommendation quality. Trust networks can be seen as a subconcept of the more general, social network based relation, which however also increases the recommendation quality [6][13][7]. Social relationships also influence the shopping behaviour of people, which mechanism relates to the strengths of weak ties in networks [3]. The difference between trust networks and symmetric social networks is somehow similar to conditional and joint probability. In the case of trust network the information (trust) is represented in a directed graph, while in the case of a social network, social relationships are represented with non-directed edges. However, asymmetric social relationships are also a useful information source [4].

3 The Graph.

An important aspect of our research is to keep the information representation method as general as possible. This way we have the opportunity to work with various calculation methods in the future. We focus on adaptive methods where we can adapt the weights belonging to different information types according to new information available. On the other side, our intention is to present as much information as possible in order not to constrain the coverage and precision of the recommendation methods.

3.1 Definition. We introduce our representation method as a labeled, weighted, restricted hypergraph, as

$$\mathcal{K} = (N, E, T_N, T_E, t_N, t_E, w_E, A, a_n, a_e)$$

. N represents the set of nodes existing in the graph, $E \subseteq \{\{u, v\} | u \in N \wedge v \in N\}$ represents the set of edges between the nodes. T_N is the set of node types, T_E is the set of edge types. Function $t_N \subset N \times T_N$ assigns a node type to each node, function $t_E \subset E \times T_E$ assigns an edge type to each edge. Function $w_E \subset E \times \mathbf{R}$ assigns weights to the edges. A is a set containing attribute values, which can be assigned to nodes or edges. Function $a_n \subset N \times A$ assigns attribute values to nodes, function $a_e \subset E \times A$ assigns attribute values to edges.

Node types define the type of entities represented in the knowledge base. Edge types define the different kinds of relations between the entities. This way the knowledge base is designed to be flexible to hold information types defined by the application domain. The intention behind this representation method is to define a framework to represent the information and provide calculation methods, guidelines and methodology for concrete applications. Edge weights (w_E) let the application assign weights to relations. To provide an example, the strength of a friendship relation (close friends, distant friends) in a concrete application scenario can be represented utilizing edge weights. Another example is item similarity, which also can be represented as the weights of the edges. Attributes have been intro-

duced to let applications assign additional information to nodes and edges. Such an item of information represented as attribute value is for instance the rating value of an edge which represents user rating value over a specific item.

3.2 MovieLens. As the method introduced in Section 4.1 is capable to incorporate heterogeneous information source types, we decided to use the MovieLens [8] 1M dataset, which we have found relatively rich in user and item attributes. MovieLens 1M is published by Grouplens[1]. The rating data also contain a time stamp, which is important for the evaluation of our methods. The dataset contains 1 000 209 anonymous ratings of 3 883 movies made by 6 040 MovieLens users who joined MovieLens in 2 000.

MovieLens 1M contains three data files in a proprietary tabular format. The data files hold user, item and rating data. The user data file contains user attributes about each user, as user id, gender, age, occupation and zip-code. The item data file contains item attributes about each movie, as movie id, title and list of genres. The ratings data file contains the user ratings on items, as user id, movie id, rating and time stamp.

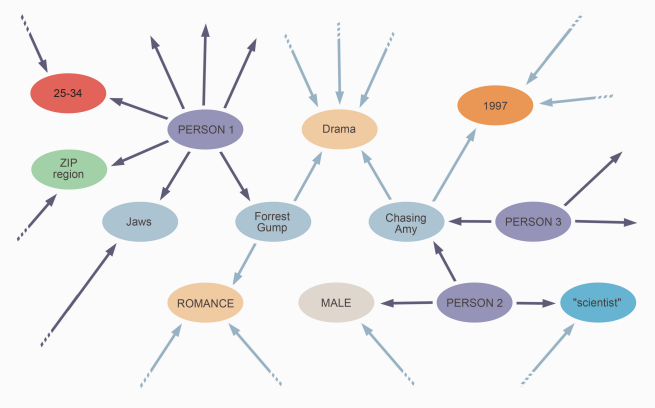


Figure 1: A detailed view of the MovieLens database represented in a directed graph

The MovieLens 1M data is represented with the knowledge base introduced in Section 3.1. As illustrated on Figure 1, the following node types are introduced to represent entities `Person`, `Item`, `AgeCategory`, `Gender`, `Occupation`, `ZipCodeRegion`, `Genre` and `YearOfPublishing`. Nodes of type `Person` represent users, nodes of type `Item` represent movies. User attribute values are represented with nodes. Node type `AgeCategory` is used to represent age category, `Gender` to represent gender, `Occupation` to represent occupation and `ZipCodeRegion` to represent U.S. region. The type `ZipCodeRegion` is a calculated attribute value. The original dataset contains U.S. zip codes. The first digit of the U.S. zip code system represents postal region, which information has been encoded into the attribute nodes. Item attributes are represented similarly. Nodes of type `Genre` represent item genre, nodes of type `YearOfPublishing` represent different years when particular movies were published.

The following edge types have been introduced to represent relations between the entities in the knowledge base `PersonAgeCategory`, `PersonGender`, `PersonOccupation`, `PersonZipCodeRegion`. `ItemGenre`, `ItemYearOfPublishing` and `ItemRating`. Edge types starting with `Person` represent relations between users and user attribute nodes representing user attribute values. For instance edges of type `PersonAgeCategory` represent the information revealing that a person belongs to an age category, `PersonGender` represents the gender of a person, `PersonOccupation` represents the occupation of a person, `PersonZipCodeRegion` represents the place in U.S. region where a person lives. Edge types starting with `Item` except `ItemRating` represent relations between items and nodes representing item attribute values. Edges of type `ItemGenre` represent the information showing which specific genre a movie belongs to, `ItemYearOfPublishing` represents that a movie was published in a specific year.

Edges of type `ItemRating` existing between users and items represent that a user rated an item with a specific value of rating. In this case the rating value was assigned to the edge as an edge attribute. The edges of this type have a special, additional attribute, the rating value. During the import process the rating values are transformed to the $[0.2, 1]$ real interval from the $[1, 5]$ integer interval by dividing the rating values by 5.

4 Recommendation.

Cold start is a common and frequently mentioned problem of recommender systems. Collaborative filtering has no reliable information to derive recommendations from for a newcoming user because the user did not express interest in a sufficient number of items. Content based methods also need information to be able to model the taste of the user. Based on this information, relevant items can be recommended. Knowledge based methods are a possible solution for this problem but their drawback is that these methods in most cases also require user interaction. Our objective is to start recommending relevant items as early as possible. To accomplish this, the highest possible amount of information is rep-

[1] `http://www.grouplens.org/`

resented and a calculation method is defined, which has the ability to derive useful information from heterogeneous data. This strategy ensures high coverage. We define the coverage as the percentage of the cases the recommendation method was able to provide a rating estimation until the corresponding step.

We introduce a spreading activation [15] based technique, which – because of the similar technique – we call recommendation spreading. Spreading activation is a well-known method in the field of semantic networks, neural networks and associative networks [1]. By utilizing spreading activation, our calculation method is able to combine different paths between the source node (i.e. the person we are generating recommendations for) and the recommended nodes (i.e. the items we are recommending). The method is sensitive to the length of each path, in order to downgrade the influence of nodes topographically far from the node in question, i.e. the node the recommendations are generated for.

4.1 Spreading. Recommendation spreading is an iterative method. In the initial step, the activation of the source nodes, is set to a constant value, in most cases to 1. The source nodes are the nodes the recommendations are generated for. In the simplest case the set of source nodes consists of one node, the node representing a person to generate the recommendations for.[2] In each step for each node a part of the activation is distributed to the neighbouring nodes, another part is kept at the activating node. The former parameter, which determines the amount of distributed activation is called `spreading relax`. The latter parameter, which determines the amount of activation kept at the node is named `activation relax`. Both parameters are real numbers and are global settings for the whole network. The outgoing activation is divided along the outgoing edges based on the weight of the edges. All these spreading values are summed up at each receiving node. This way the sum of the activations received by the destination nodes is the same amount as the activations distributed from the source node. If it's important to keep the sum of the activations at a constant level in the network, the sum of activation relax and spreading relax must be equal to one.[3] The concrete spreading relax and activation relax values are the parameters of our spreading method.

Another feature of our spreading methods is the threshold constant. If the activation of a node falls below the threshold constant, its activation will be set to zero [2]. The threshold is introduced to avoid unnecessary computation of activations close to zero. This parameter is called `activation threshold`.

There are various options to define the termination criteria for recommendation spreading. A relatively simple solution is the one based on iteration step, i.e. to stop the iteration after a certain iteration step count is reached. A delta based termination criterion could also be used meaning to run the iteration until there is no significant change in the activations [17]. We decided to use a step limit, because we think that this termination criterion fits better to our intended purpose. In a recommendation scenario a method is necessary, which delivers nodes topographically close to the source nodes. It is also important to mention that calculating a delta based termination criterion is resource intensive, while step limit is cheap to compute. A limit on iteration steps has been introduced, which parameter is called `step limit`.

4.2 Network Based Comparison. In the following we are comparing collaborative filtering with our approach.

Figure 2 illustrates a simplified collaborative filtering scenario. Nodes labeled with "U" represent users, nodes labeled with "I" indicate items. Edges labeled with "R" represent ratings. In this sample scenario we are generating rating estimation for user "U1". User "U1" rated two items, "I1" and "I2" in common with user "U2". User "U1" rated item "I3" in common with user "U3". The rating estimation for item "I4" is calculated as a weighted sum of rating "R7" and "R8". The weights for rating "R7" are influenced by dashed edges, namely "R1", "R2", "R4" and "R5", the weights for rating "R8" are influenced by dotted edges, "R3" and "R6".

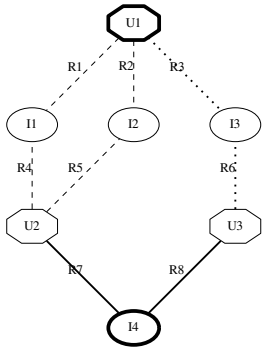


Figure 2: Collaborative Filtering

Figure 3 illustrates our approach, where we show how heterogeneous information can be incorporated to

[2]It is possible to start the recommendation spreading from multiple nodes. For example if a user is browsing an item, then the spreading can be started from the two nodes representing the user and the item. This way the final recommendation result can be influenced by both the user and the browsed item.

[3]Setting the weights to appropriate values won't prevent spreading methods to lose activation. In a case of a node with no outgoing edges the node cannot redistribute its activation to any neighbouring node, thus its activation will be lost. A possible solution to overcome this problem is to bind all nodes to the source node with a backlink edge. This method has other advantages as discussed by Zielger et al. [17], called avoidance of dead ends.

produce rating estimation. Nodes labeled with "U" represent users, nodes labeled with "I" indicate items, the node labeled with "A" represents age category. Age category is an example how to represent user attributes. Item attributes can be represented similarly. Edges labeled with "R" represent ratings, edges labeled with "A" mean user belongs to an age category, edge labeled with "S" means item similarity. User "U1" rated item "I1" and "I2" in common with user "U2". The dashed edges (R1, R2, R4, R5) represent the influence of user similarity on the weight of rating "R7" on final estimation. User "U1" belongs to the same age group "A" with user "U3". The solid edges (A1, A2) represent the same age group relation and their influence to the weight of rating "R8". It also illustrates how user and item attributes can influence the final recommendation result. The dotted path (R3,S1,R6) starts from user "U1" with a rating edge, continues with an edge representing item similarity and ends with a rating edge. The semantics behind this path could be described as user "U1" and user "U4" rated a similar item. Activation received through the dotted path specifies the weight of rating "R9".

4.3 Rating Weights. The introduced recommendation method can also be treated as the generalization of collaborative filtering in the case of representation of heterogeneous information. While calculating recommendation spreading, the activation flowed through each rating edge is accumulated. We denote this value with a, which variable is indexed with the edge. To calculate the weighted sum of estimated rating, these accumulated values are used as the weights of rating values belonging to the specific edge. Recommendation spreading calculates rating estimations with the following formula

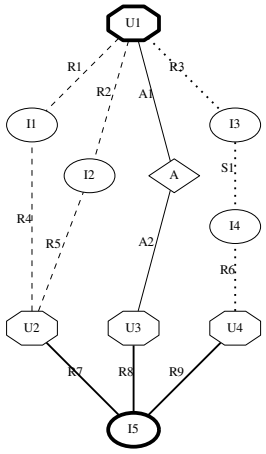


Figure 3: Recommendation Spreading

$$\hat{r}_{u,i} = \bar{r}_u + \frac{\sum_{e \in E_{ItemRating} \wedge i \in e \wedge v \in e \wedge i \neq v} (r_{v,i} - \bar{r_v}) a_e}{\sum_{e \in E_{ItemRating} \wedge i \in e \wedge v \in e \wedge i \neq v} a_e}$$

, where $\hat{r}_{u,i}$ is the estimated rating value for user u on item i, $\bar{r}_u$ and $\bar{r_v}$ denote the average of the already issued ratings by user u and v respectively, $r_{v,i}$ is the known rating value of user v on item i, a_e is the aggregated activation flowed through via edge e, $E_{ItemRating} = \{e \in E | t_E(e) = \texttt{ItemRating}\}$ is the set of edges of type `ItemRating`.

Rating edges are drawn between users and items. This means that in the sample case shown in Figure 3 user similarity between `U1` and `U2` can be defined as the activation arrived from `U1` to `U2` through the network. [4]

4.4 Flow direction. The knowledge base represents the information with a weighted directed graph. Figure 3 shows a sample spreading scenario. If "Item 5" would be recommended to "User 1", a directed path between these nodes couldn't be found. With a recommendation spreading method flowing only in the direction of the edges coverage would be lost. Spreading on an undirected graph also means that if a node received an activation in step i, in step $i+1$ it will spread a part of its activation back to the node it received activation from through the previously receiving edge.

5 Evaluation.

5.1 Evaluation Method. We call our evaluation method time series evaluation, which is an iterative method based on time stamped data. The evaluation iterates through time stamped data of sample items in ascending order, while repeating the following operations

1. take the next rating sample from the database,
2. ask for rating estimation from the recommender engine,
3. calculate the rating error and record it to the evaluation log,
4. add the true rating value to the knowledge base as an edge of type `ItemRating`.

Before starting the evaluation, the knowledge base is filled with all the information found in the MovieLens 1M dataset except the `ItemRating` edges. We decided to use this method to simulate real life scenarios, with a focus on the ability of comparing methods in the beginning, cold start steps, when there is only a low amount of rating information available in the knowledge base. The knowledge base is filled with additional information (`ItemRating` edges) during the evaluation process.

5.2 Numerical Experiment. We were interested in comparing different `step limit` settings to collaborative filtering. Table 1 summarizes the methods which were evaluated in the experiment. For easier reference

[4] The statement holds, because `U2` has only one outgoing rating edge. The activation leaves from `U2` only via `R7`.

Name	Method	Method parameters
CF	Collaborative Filtering	–
S3	Recommendation Spreading	Step limit: 3
S4	Recommendation Spreading	Step limit: 4
S5	Recommendation Spreading	Step limit: 5
S6	Recommendation Spreading	Step limit: 6
S7	Recommendation Spreading	Step limit: 7
S8	Recommendation Spreading	Step limit: 8

Table 1: Engines and configurations

later we assigned a name to these methods which can be found in the first column of the table. We evaluated the recommendation spreading method on the MovieLens 1M dataset represented as described in Section 3.2. Time series evaluation method was conducted as described in Section 5.1. As we were interested in how the methods perform in the information sparse environment, we ran the experiment on the first 10 002 rating values of the sample dataset. The benchmark method in the experiments is collaborative filtering with a Pearson correlation based similarity calculation method. The `activation relax` parameter of the spreading method has been set to 0.5, the `spreading relax` has been also set to 0.5. The `activation threshold` of spreading methods has been set to 0, meaning no thresholding, in order to see the pure, unoptimized performance of spreading methods.

5.3 Rating Estimation Error. Figure 4 compares two methods, namely S3 and CF. We decided to compare S3 with CF as S3 is the most restricted spreading method. Spreading methods running for higher step limits have a higher chance to deliver recommendation estimations of better quality. The horizontal axis of the figure represents evaluation steps, the vertical axis represents the MAE until the corresponding step. Seeing the step interval below 1 000, the figure shows that the recommendation spreading method has a higher coverage in the cold start case. Furthermore, in this region the quality of the estimated values of S3 are higher than the estimated values of the CF method, which statement holds until approximately the 3 000th step. To summarize the information on Figure 4, recommendation spreading converges faster and it has a higher coverage in the cold start case. Coverage is defined as the percentage of cases the recommendation method was able to provide a rating estimation at until the corresponding evaluation step.

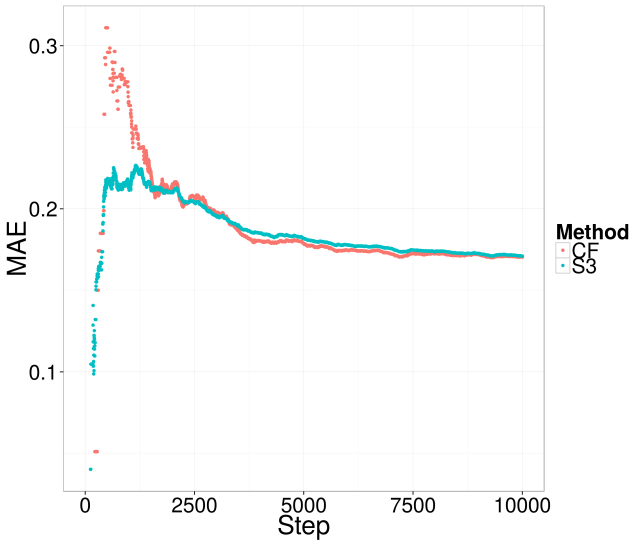


Figure 4: Comparing the MAE of recommendation spreading and collaborative filtering

5.4 Coverage. Figure 5 compares the coverage of S3 and CF methods. The horizontal axis of the figure represents evaluation steps, the vertical axis represents the coverage at the corresponding step. Figure 5 shows that recommendation spreading provides higher coverage than collaborative filtering. The difference on coverage is also high in the beginning steps, when the knowledge graph is more sparse on true rating values. It means that recommendation spreading performs better in the cold start case than collaborative filtering. We explain this by treating the coverage problem as finding a path between two nodes. While collaborative filtering can operate on a restricted set of edges (only on the `ItemRating` edges), recommendation spreading can utilize any type of edge. This is the reason the spreading based method can reach the item node in a higher number of cases.

Engine name	Coverage	MAE
CF	6 118	0,170 4
S3	7 897	0,171 0
S4	7 897	0,171 0
S5	7 910	0,171 1
S6	7 910	0,170 8
S7	7 910	0,170 7
S8	7 910	0,170 5

Table 2: Coverage and MAE of different engines at the last evaluation step

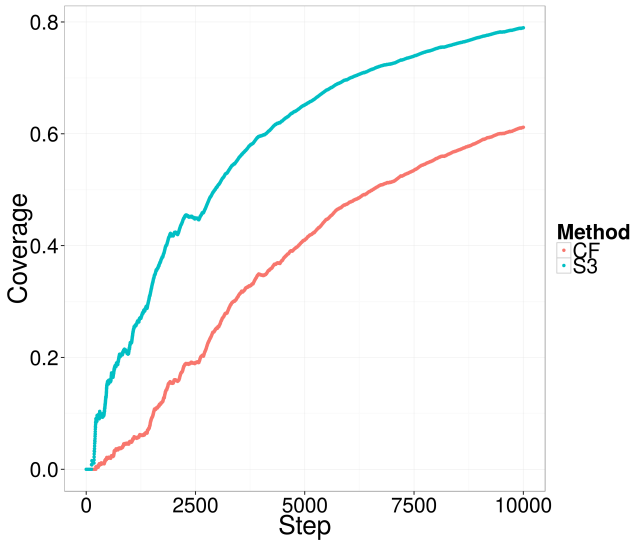


Figure 5: Comparing the coverage of recommendation spreading and collaborative filtering

Table 2 contains the coverage and MAE values of the engines in the experiment at the last evaluation step. The MAE values are very similar, differing only at the third digit. It means that recommender spreading is a method which successfully combines heterogeneous information sources with the same estimation error as collaborative filtering on the MovieLens 1M dataset. Regarding coverage, Table 2 provides two insights. As the spreading method has more options to reach the destination node from the source node, this method is able to estimate ratings in more cases. The second consequence is that a higher spreading `step limit` value does not necessarily lead to a significantly higher coverage. It means that this parameter is sensitive to the underlying data or application domain and should be fine tuned for each dataset or application.

6 Conclusion.

The results show that a rating estimation method was developed which has very similar prediction error as collaborative filtering has. As by its nature recommendation spreading works from a higher number of rating estimations, the method has a higher coverage than the benchmark method has. Comparing recommendation spreading methods configured to different spread `step limits` shows that while the coverage increases with the step limit, the precision of rating estimation does not increase or decrease. It means if an application of the method needs an estimation on a specific item, the iteration can be stopped as spreading reached the node representing the item, because further spreading does not increase the precision.

Next to its higher coverage, an important property of recommendation spreading is its faster convergence. The results can be explained by a higher number of rating values reached to aggregate. As the results also show that the higher the number of aggregated rating values leads to the lower error of rating estimation, the faster convergence can be the consequence of the aggregation of a higher number of rating values also in the beginning, cold start case. It was also shown that in the long term, the introduced method has very similar precision as collaborative filtering has. It means that a method was developed which is able to combine a higher number of ratings while not increasing the error of rating estimation.

We would like to extend the knowledge base or the recommendation engines with an additional information, the information type weight function. Type weights express the strength of a relation type. This information can be used by the calculation method. Introducing type weights, our intention is to let the model or the calculation method store the importance of different relation types. For example the weight of the relation type representing friendship can be set to 0.8, the weight of the relation type representing country of production can be set to 0.4. The values express the importance of the various information types. The calculation methods can use this information when calculating recommendations, for instance by multiplying the relation type weight with the relation weight. The potential of introducing type weights can be found in the learning capability of the recommendation methods. If a calculation method is capable to react on user feedback, it can have a training method to adjust relation type weights according to feedbacks from the environment. Tuning these weights can lead to an increase in the recommendation quality. Manual tuning requires a domain expert and does not guarantee a quick and better result. One of our future plans is to develop a training method which adjusts the weights of the different information source types based on user feedbacks, letting the recommendation method continuously adapt to the changes in the environment.

Currently the building blocks of the knowledge base are information representation units. The graph based model represents the information with nodes and edges. To utilize the information collected in the knowledge base, recommendation spreading algorithm has been used. There are several options to process this information, for example we could also work with a random walk based method. In Section 5 we showed that recommendation spreading has the potential to have a high

coverage but the error of the rating estimation could not have been made lower. Our suspicion is that by introducing a finer grain method, it would be possible to increase the recommendation quality. One option is to utilize neural networks, thus to change information representation units – graph nodes – to artificial neurons to let the network adapt to its environment by training itself.

References

[1] Nicholas V. Findler, editor. *Associative Networks: The Representation and Use of Knowledge of Computers*. Academic Pr, 1979.

[2] Stephan Gouws, GJ Van Rooyen, and Herman A Engelbrecht. Measuring conceptual similarity by spreading activation over Wikipedia's hyperlink structure. In *Proceedings of the 2nd Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, page 46, 2010.

[3] Mark Granovetter. The Strength of Weak Ties. *The American Journal of Sociology*, 78(6):1360–1380, 1973.

[4] Hansu Gu, Mike Gartrell, Liang Zhang, Qin Lv, and Dirk Grunwald. AnchorMF: Towards Effective Event Context Identification. In *Proceedings of the 22Nd ACM International Conference on Conference on Information & Knowledge Management*, CIKM '13, pages 629–638, New York, NY, USA, 2013. ACM.

[5] R. Guha, Ravi Kumar, Prabhakar Raghavan, and Andrew Tomkins. Propagation of trust and distrust. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 403–412, New York, NY, USA, 2004. ACM.

[6] Ido Guy, Naama Zwerdling, David Carmel, Inbal Ronen, Erel Uziel, Sivan Yogev, and Shila Ofek-Koifman. Personalized recommendation of social software items based on social relations. In Lawrence D. Bergman, Alexander Tuzhilin, Robin D. Burke, Alexander Felfernig, and Lars Schmidt-Thieme, editors, *RecSys*, pages 53–60. ACM, 2009.

[7] Jianming He. *A Social Network-based Recommender System*. PhD thesis, Los Angeles, CA, USA, 2010. AAI3437557.

[8] J.L. Herlocker, J.A. Konstan, A. Borchers, and J. Riedl. An Algorithmic Framework for Performing Collaborative Filtering. In *Proceedings of International ACM Conference on Research and Development in Information Retrieval (SIGIR)*, pages 230–237, Berkeley, USA, August 1999.

[9] Balázs Hidasi and Domonkos Tikk. Fast ALS-Based Tensor Factorization for Context-Aware Recommendation from Implicit Feedback. In Peter A. Flach, Tijl De Bie, and Nello Cristianini, editors, *ECML/PKDD (2)*, volume 7524 of *Lecture Notes in Computer Science*, pages 67–82. Springer, 2012.

[10] Tim Hussein and Sebastian Neuhaus. Explanation of Spreading Activation Based Recommendations. In *Proceedings of the 1st International Workshop on Semantic Models for Adaptive Interactive Systems*, SEMAIS '10, pages 24–28, New York, NY, USA, 2010. ACM.

[11] Audun Jøsang, Stephen Marsh, and Simon Pope. Exploring Different Types of Trust Propagation. In Ketil Stølen, William H. Winsborough, Fabio Martinelli, and Fabio Massacci, editors, *iTrust*, volume 3986 of *Lecture Notes in Computer Science*, pages 179–192. Springer, 2006.

[12] P. Kazienko, K. Musial, and T. Kajdanowicz. Multidimensional Social Network in the Social Recommender System. *Trans. Sys. Man Cyber. Part A*, 41(4):746–759, July 2011.

[13] Ioannis Konstas, Vassilios Stathopoulos, and Joemon M. Jose. On Social Networks and Collaborative Recommendation. In *Proceedings of the 32Nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, pages 195–202, New York, NY, USA, 2009. ACM.

[14] Paolo Massa and Paolo Avesani. Trust-Aware Collaborative Filtering for Recommender Systems. In Robert Meersman and Zahir Tari, editors, *CoopIS/DOA/ODBASE (1)*, volume 3290 of *Lecture Notes in Computer Science*, pages 492–508. Springer, 2004.

[15] M. Ross Quillian. Semantic memory. In Marvin Minsky, editor, *Semantic Information Processing*, pages 227–270. MIT Press, Cambridge, MA, 1968.

[16] Paul Resnick, Neophytos Iacovou, Mitesh Suchak, Peter Bergstrom, and John Riedl. GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In *Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work*, CSCW '94, pages 175–186, New York, NY, USA, 1994. ACM.

[17] Cai-Nicolas Ziegler and Georg Lausen. Propagation Models for Trust and Distrust in Social Networks. *Information Systems Frontiers*, 7(4-5):337–358, December 2005.

Decentralized Recommender Systems

Zhangyang Wang* Xianming Liu* Shiyu Chang* Jiayu Zhou† Guo-Jun Qi‡
Thomas S. Huang*

Abstract

This paper proposes a *decentralized recommender system* by formulating the popular collaborative filleting (CF) model into a decentralized matrix completion form over a set of users. In such a way, data storages and computations are fully distributed. Each user could exchange limited information with its local neighborhood, and thus it avoids the centralized fusion. Advantages of the proposed system include a protection on user privacy, as well as better scalability and robustness. We compare our proposed algorithm with several state-of-the-art algorithms on the FlickerUserFavor dataset, and demonstrate that the decentralized algorithm can gain a competitive performance to others.

1 Introduction

The paper discusses the decentralized recommender systems, which is in contrast to the typical recommender systems built on centralized infrastructures (the "cloud", etc.). The decentralized network [1] had been thoroughly investigated in control and communication fields, defined as a set of distributed but connected agents, who are generally not strongly connected in a graph theoretic sense. Each agent collects data by itself, and executes computation via limited communication within only local neighborhoods. Fig. 1 shows a comparison of centralized versus decentralized network structures. Specifically, in a decentralized recommender system, individual users / user-groups can be viewed as network agents. Each user holds his or her own ratings as partially observed data. The data cannot be accessed by any intermediate point or centralized server in the network. Therefore, it has a potential effect on protecting user data privacy against both the cloud server and some malicious eavesdropping over uploading channels. For a large-scale network of mobile users, the decentralized models own a better scalability since users are only locally connected. Since data storages and computations are fully distributed, the decentralized systems also become robust to center (cloud) or individual agent (user) failures.

To our best knowledge, we are the first studying and designing a decentralized recommender system. There have been some recent interests in investigating algorithms in a decentralized fashion [2, 3], and moreover, preliminary literatures to reveal the convergence [4] and dynamics [5] properties. All above make it solid and promising to build application scenes on a decentralized network structure.

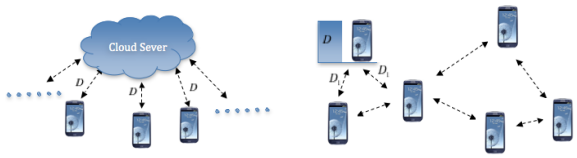


Figure 1: The comparison of a centralized network (left) and a decentralized network (right).

2 Model and Algorithm

As a most popular tool in recommender system, Collaborative Filtering (CF) is usually formulated as matrix factorizations problem [7] [11]. It predicts user rating $\mathbf{R}_{i,j}$ (of i-th user on j-th item) as a dot product of the user profile of the i-th user, denoted as a row vector $\mathbf{U}^i$, and the item profile of the j-th item denoted as a column vector $\mathbf{V}_j$, i.e., $\mathbf{R}_{i,j} = \mathbf{U}^i\mathbf{V}_j$. The recommendation problem can be formulated as solving the following matrix factorization problem:

$$\begin{aligned} &\min_{\mathbf{U},\mathbf{V},\mathbf{Z}} \tfrac{1}{2}||\mathbf{UV}-\mathbf{Z}||_2^2 \\ &s.t. \quad P_\Omega(\mathbf{Z}) = P_\Omega(\mathbf{R}) \end{aligned} \tag{2.1}$$

Here P_Ω denotes the projection over the set of available ratings, and $\mathbf{Z}$ is an auxiliary matrix.

It is assumed that CF is performed by L users jointly in a decentralized manner, and $\mathbf{R}$ is segmented into L non-overlapped parts, denoted as $\mathbf{R}_i$, $i = 1, 2, ..., L$. For example, the easiest case to segment $\mathbf{R}$ is to divide by columns. The i-th user ($i = 1, 2, ..., L$) observes $\mathbf{R}_i$. Note some level of synchronization is still required to collaboratively utilize information from all users. The trade-off strategy is **to share partial data only among users in the local neighborhood**. After observing the problem structure, authors in [2] suggested an variant of nonlinear Gauss-Seidel (GS) iterations, named *decentralized matrix completion* (**DMC**) algorithm. The i-th user will hold $\mathbf{R}_i$, as well as $\mathbf{U}_i$, $\mathbf{V}_i$, and $\mathbf{Z}_i$ based on its own computations. Note $\mathbf{U}_i$ here is of the same size as $\mathbf{U}$, and $\mathbf{Z}_i$ of the same dimension as $\mathbf{R}_i$, so in other words,

*Beckman Institute, University of Illinois at Urbana-Champaign, IL 61821. {zwang119, xliu102, chang87, t-huang1}@illinois.edu.
†Samsung Research America, San Jose, CA 95134. jiayu.zhou@samsung.com
‡University of Central Florida, Orlando, FL, 32816, USA. guojun.qi@ucf.edu.

Algorithm 1 Decentralized matrix completion (DMC) algorithm for solving (2.1)

Require: $P_\Omega(\mathbf{R}_i), (i = 1, 2, ..., L)$; initializations of $\mathbf{U}_i$, $\mathbf{V}_i$, and $\mathbf{Z}_i$ for each i-th user ($i = 1, 2, ..., L$); step size β; ITER

1: FOR t=1 to ITER DO
2: Each i-th user updates $\mathbf{V}_i$: $\mathbf{V}_i = (\mathbf{U_i^T U_i})^{-1}\mathbf{U_i^T Z_i}$
3: Each i-th user updates $\mathbf{Z}_i$: $\mathbf{Z}_i = \mathbf{U_i V_i} + P_\Omega(\mathbf{R_i} - \mathbf{U_i V_i})$
4: Each i-th user propagates $\mathbf{U}_i$ to its one-hop neighborhood N_i.
5: Each i-th user updates $\mathbf{U}_i$:
$\mathbf{U}_i = \frac{\mathbf{Z}_i\mathbf{V}_i^T - \mathbf{a}_i + \beta\sum_{j\in N_i}\mathbf{U}_j}{1+2\beta|N_i|}$
$\mathbf{a}_i = \mathbf{a}_i + \beta(|N_i|\mathbf{U}_i - \sum_{j\in N_i}\mathbf{U}_j$
6: END

Ensure: $\mathbf{U}_i$, $\mathbf{V}_i$, and $\mathbf{Z}_i$, $i = 1, 2, ..., L$

$\mathbf{Z}_i = \mathbf{U}_i\mathbf{V}_i$. In each iteration, the i-th user first updates $\mathbf{V}_i$ and $\mathbf{Z}_i$ independently, then exchanging $\mathbf{U}_i$ with its one-hop connected neighborhood users, and finally update $\mathbf{U}_i$ via the average consensus algorithm[6]. The algorithm is summarized in Algorithm I. It obtains similar reconstruction errors, compared to centralized solutions [2].

3 Experiments

We compare our proposed algorithm with state-of-the-arts in this section, including Probabilistic Matrix Factorization (*PMF*) [8], and Collaborative Topic Modeling (*CTR*) [9], on a collected image recommendation dataset from Flickr.

FlickrUserFavor dataset: The dataset contains $350,000$ images collected from Flickr, from 140 user groups and uploaded by 20,298 users. We use the *"like"* feedback provided by users as binary ratings. 75% of the rating matrix is used as training, and 25% as testing.

Evaluation Measurement: We use the averaged ranked order of all the rated images in the testing dataset for a specific user to evaluate performances. Among these ranked images, we determine those for which a user has exhibited a *"like"* preference in the *test data*, and report the *average percentile score* (APS) of the ranked images which are indeed preferred by the user. The lower the APS, the better the algorithm is, which means the user preferred images are ranked in top positions. Finally, the *mAPS* is reported with the mean of the APS scores for all target users.

Performance Comparision: PMF [8] is the most classical collaborative filtering algorithm for recommender system. Wang et.al. also proposed the Collaborative Topic Model [9] which involves both content and user ratings, where we use the Hierarchical Gaussianization (HG) [10] as the image features. For the proposed decentralized algorithm, we set the rank as 64, and using 8 agents. Detailed *mAPS* comparisons are shown in Table 1. It is shown that DMC is capable to achieve competitive performance as *CTR*, while is far better than *PMF*. Moreover, we do not use any content information in our algorithm, while *CTR* uses content to indicates similarities between items. That suggests a further direction improve our algorithm too.

Table 1: Performances of the proposed approaches compared with other baseline methods. The second column indicates whether or not the algorithm uses content information. Without using content information and the fusion center, the proposed algorithm achieves a competitive performance.

method	Content	*mAPS*
PMF [8]	N	61.99
CTR [9]	Y	52.76
DMC	N	53.46

4 Conclusion

This paper discusses a decentralized recommender system. We formulate the popular collaborative filleting model into a decentralized matrix completion problem. Each user, with only partial rating data, can exchange user profile factors with its local neighborhood, while keep item profile factors private. We compare our proposed algorithm with several state-of-the-arts on the FlickerUserFavor dataset, and illustrate comparable results to the conventional ones.

5 Acknowledgement

The work is supported by Samsung under the 2013 GRO project "Device centric Social Network Platform".

References

[1] C. Yeung, I. Liccardi , K. Lu , O. Seneviratne and T. Lee, "Decentralization: The future of online social networking", In W3C Workshop on Future of Social Networking Position Papers.
[2] Q. Ling, Y. Xu, W. Yin and Z. Wen, "Decentralized low-rank matrix completion," in Proceedings of ICASSP , pp. 2925-2928, 2012.
[3] Q. Ling, Z. Wen, W. Yin, "Decentralized jointly sparse optimization by reweighted ℓ_q minimization," IEEE Trans. on Signal Processing, 61.5 (2013): 1165-1170.
[4] W. Shi, Q. Ling, K. Yuan, G. Wu, and W. Yin, "On the linear convergence of the ADMM in decentralized consensus optimization," (2014): 1-1.
[5] Q. Ling and A. Ribeiro, "Decentralized dynamic optimization through the alternating direction method of multipliers," IEEE 14th Workshop on SPAWC, pp. 170-174, 2014.
[6] T. Erseghe, D. Zennaro, E. DallAnese, and L. Vangelista, "Fast consensus by the alternating direction multipliers method," IEEE Trans. on Signal Processing, 59.11 (2011): 5523-5537.
[7] Y. Koren, R. Bell, and C. Volinsky, Matrix factorization techniques for recommender systems, Computer, vol. 42, no. 8, pp. 30-37, 2009.
[8] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in Proceedings of *NIPS*, 2008, vol. 20.
[9] C. Wang and D. M. Blei, "Collaborative topic modeling for recommending scientific articles," in Proceedings of *SIGKDD*, 2011, pp. 448 – 456.
[10] X. Zhou, N. Cui, Z. Li, F. Liang, and T. S Huang, "Hierarchical gaussianization for image classification," in Proceedings of *ICCV*. pp. 1971–1977, 2009
[11] S. Chang, G. Qi, C. Aggarwal, J. Zhou, M. Wang, and T. S. Huang, Factorized similarity learning in networks. in Proceedings of *ICDM*, 2014.