PROBABILISTIC LATENT SEMANTIC ANALYSIS BASED FRAMEWORK FOR HYBRID SOCIAL RECOMMENDER SYSTEMS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ERKİN ERYOL

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
COMPUTER ENGINEERING

MAY 2010

Approval of the thesis:

PROBABILISTIC LATENT SEMANTIC ANALYSIS BASED FRAMEWORK FOR HYBRID SOCIAL RECOMMENDER SYSTEMS

submitted by **ERKİN ERYOL** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen _____
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Adnan Yazıcı _____
Head of Department, **Computer Engineering Dept., METU**

Assoc. Prof. Dr. Nihan Kesim Çiçekli _____
Supervisor, **Computer Engineering Dept., METU**

Assoc. Prof. Dr. Ferda Nur Alpaslan _____
Co-supervisor, **Computer Engineering Dept., METU**

Examining Committee Members:

Asst. Prof. Dr. Tolga Can
Computer Engineering Dept., METU _____

Assoc. Prof. Dr. Nihan Kesim Çiçekli
Computer Engineering Dept., METU _____

Asst. Prof. Dr. İlkay Ulusoy
Electrical and Electronics Engineering Dept., METU _____

Dr. Ayşenur Birtürk
Computer Engineering Dept., METU _____

M.Sc. Özgür Alan
ORBIM Software _____

Date: 03.05.2010

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Erkin ERYOL

Signature :

ABSTRACT

PROBABILISTIC LATENT SEMANTIC ANALYSIS BASED FRAMEWORK FOR HYBRID SOCIAL RECOMMENDER SYSTEMS

Eryol, Erkin

M.S., Department of Computer Engineering

Supervisor: Assoc. Prof. Dr. Nihan Kesim Çiçekli

Co-Supervisor: Assoc. Prof. Dr. Ferda Nur Alpaslan

May 2010, 78 pages

Today, there are user annotated internet sites, user interaction logs, online user communities which are valuable sources of information concerning the personalized recommendation problem. In the literature, hybrid social recommender systems have been proposed to reduce the sparsity of the usage data by integrating the user related information sources together. In this thesis, a method based on probabilistic latent semantic analysis is used as a framework for a hybrid social recommendation system. Different data hybridization approaches on probabilistic latent semantic analysis are experimented. Based on this flexible probabilistic model, network regularization and model blending approaches are applied on probabilistic latent semantic analysis model as a solution for social trust network usage throughout the collaborative filtering process. The proposed model has outperformed the baseline methods in our experiments. As a result of the research, it is shown that the proposed methods successfully model the rating and social trust data together in a theoretically principled way.

Keywords: Recommender Systems, Social Trust Network, Probabilistic Latent Semantic Analysis

ÖZ

MELEZ SOSYAL TAVSİYE SİSTEMLERİ İÇİN OLASILIKSAL SAKLI ANLAM ANALİZİ TABANLI BİR ÇATI

Eryol, Erkin

Yüksek Lisans, Bilgisayar Mühendisliği Bölümü

Tez Yöneticisi: Doç. Dr. Nihan Kesim Çiçekli

Ortak Tez Yöneticisi: Doç. Dr. Ferda Nur Alpaslan

Mayıs 2010, 78 sayfa

Günümüzde kişiselleştirilmiş tavsiye probleminin çözümüne yönelik olarak kullanıcılar tarafından etiketlenen internet siteleri, kullanıcı etkileşim günceleri, çevrimiçi kullanıcı toplulukları gibi değerli bilgi kaynakları bulunmaktadır. Literatürde, melez sosyal tavsiye sistemleri, kullanıcı ile ilgili bilgi kaynaklarının bir arada kullanımı ile kullanım verisi seyrekliğini düşürmek üzere önerilmektedir. Bu tez kapsamında, melez sosyal tavsiye problemi için çatı olarak, sağlam istatistiksel temel sağlayan olasılıksal saklı anlam analizi yöntemi kullanılır. Farklı veri melezleştirme yaklaşımları üzerinden deneyler hazırlanmıştır. Bu esnek olasılıksal model üzerinde ağ düzenleme ve model harmanlama yaklaşımları sosyal güven ağının kolektif filtreleme sürecinde kullanımı için önerilmiştir. Deneylerde, önerilen yöntemler başarılı bir şekilde temel seviye yöntemlerden daha yüksek başarı göstermiştir. Araştırma sonucunda, önerilen yöntemlerin oy ve sosyal güven ağı verilerini teoriye uygun olarak bir arada modellediği gösterilmiştir.

Anahtar Kelimeler: Tavsiye sistemleri, Sosyal Güven Ağı, Olasılıksal Saklı Anlam Analizi

To my family

ACKNOWLEDGEMENTS

I would like to thank to my supervisor Assoc. Prof. Dr. Nihan Kesim Çiçekli for her patience and support. I want to thank to Asst. Prof. İlkay Ulusoy for her criticism, guidance and the time she invested in this thesis. I also want to thank my co-supervisor Assoc. Prof. Dr. Ferda Nur Alpaslan for her support, advices and guidance.

I am deeply grateful to my family for their love and support. Without them, this work could have never been completed.

I would like to thank the Scientific and Technological Research Council of Turkey (TÜBİTAK) for financially supporting this thesis under project 107E234.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

PLSA	Probabilistic latent semantic analysis
GPLSA	Gaussian probabilistic latent semantic analysis
EM	Expectation maximization
TN	Trust network
CF	Collaborative filtering
MAE	Mean absolute error
PCC	Pearson correlation coefficient
TBF	Trust Based Filtering

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

The tremendous growth rate, highly unstructured nature and sparseness of data on the World Wide Web lead to the need of devising new ways to reach relevant information. Concerning these drawbacks, modeling the data on the web is a great challenge. In parallel with the growth of data on the web, the user interaction has also increased which enables a user specific perspective for modeling the data on the web. User annotated internet sites, user interaction histories, online user communities are valuable sources of information concerning the creation of user specific perspectives. In the literature, hybrid recommender systems have been proposed to deal with the sparsity problem and to integrate various information sources together.

The main approaches on modeling the web data is based on an oversimplifying assumption which is the popularity of websites. This assumption ignores the diverse user specific needs concerning the web data model. Hence, with more concrete and accurate assumptions, web data should be modeled so that the model has a different perspective for each user. This approach is referred to as *personalization*.

Recommender systems research area is a sub-topic of the personalization problem. Recommender systems are attracting more researchers from both academia and industry every day. The Netflix competition [57] and the Recommender Systems conference RecSys [58] are such examples of this interest. Netflix competition is a recommendation system contest where contestants compete with Netflix's own recommendation system, Cinematch [57] and need to improve the root mean squared error of Cinematch by 10% for a prize of $1000000. This improvement is achieved by the team BellKor's Pragmatic Chaos and their methods are publicly reachable [59].

Recently, the social network web sites have gained significant attention and popularity. The social networks are shown to be important sources of information. The commercial survey, Nielsen trust and advertising global report [8], shows that 9 out of 10 people obey their friends' recommendations and social networks brings such an important information to online systems. Another aspect of recommender systems is the type of the recommendation algorithm. Collaborative filtering is shown to be the most accurate technique compared to other techniques of recommendation [80]. As a model based method, probabilistic latent semantic analysis provides a means of integrating several data sources together in a principled way. Expectation maximization based model training of PLSA provides adjustable trade off between the accuracy and time complexity of the model. Considering the importance of social network analysis and the advantages of probabilistic latent semantic analysis, we aim at incorporating social networks with collaborative filtering.

Our main contribution is based on the application of a recent theoretical advancement on exploiting similarity networks in the probabilistic dyadic data modelling process which is explained in Chapter 4. This application uses social trust networks in the collaborative filtering process. To the best of our knowledge, this is the first work that proposes a model based algorithm that integrates a social network throughout the collaborative filtering process on explicit ratings. This contribution is important since the method can modularly be applied on several other probabilistic recommendation models. We demonstrate that this contribution is applicable on Gaussian PLSA and build our approaches based on Gaussian PLSA. The proposed network regularization method is further improved by indirect trust based methods, TidalTrust, MoleTrust and Trust Based Filtering.

Our contributions can be listed as follows.

1. We propose a novel hybrid method for social recommendation.
2. We propose a method for network regularization on Gaussian PLSA collaborative filtering algorithm.
3. We experiment popular trust based methods as the social trust heuristics of network regularization method on Gaussian PLSA.

The thesis organization is visualized in Figure 1. Level 1 contains the background information about the methods that are presented in Chapter 2. Level 2 contains the related work topics. Level 3 lists the methods that our proposed algorithms are based on. These methods are explained in Chapter 4. The leaf nodes in the figure are the experiments of our methods. The methods presented in dotted bordered oval boxes are original to this work.

The thesis is organized as follows. Chapter 2 presents the fundamental information on personalization and recommender systems. The terminology and the details regarding the input data and our solution to the recommendation are also explained in this chapter.

In Chapter 3, the related work is presented. The first part of this chapter is dedicated to the overview of recommender systems. The rest of this chapter is related to variations of the PLSA model. First, the pairwise contraint integration to PLSA model training is explained. Then, PLSA based recommendation models which are previously proposed in the literature are explained.

Chapter 4 contains the main contribution of this thesis, which is integrating social networks in the Gaussian PLSA model. First, the details of Gaussian PLSA is given. The other approach that we adopt is network regularization on PLSA model. The network regularization approach is explained in the next part. This part includes the network regularization method on PLSA model for document clustering and the contribution of this thesis, which is the application of network regularization on Gaussian PLSA. Another contribution of this work is the usage on indirect trust based methods as the core of network regularization. The indirect trust based methods and their usage for trust enhanced collaborative filtering (CF) problem are in the last part of this chapter.

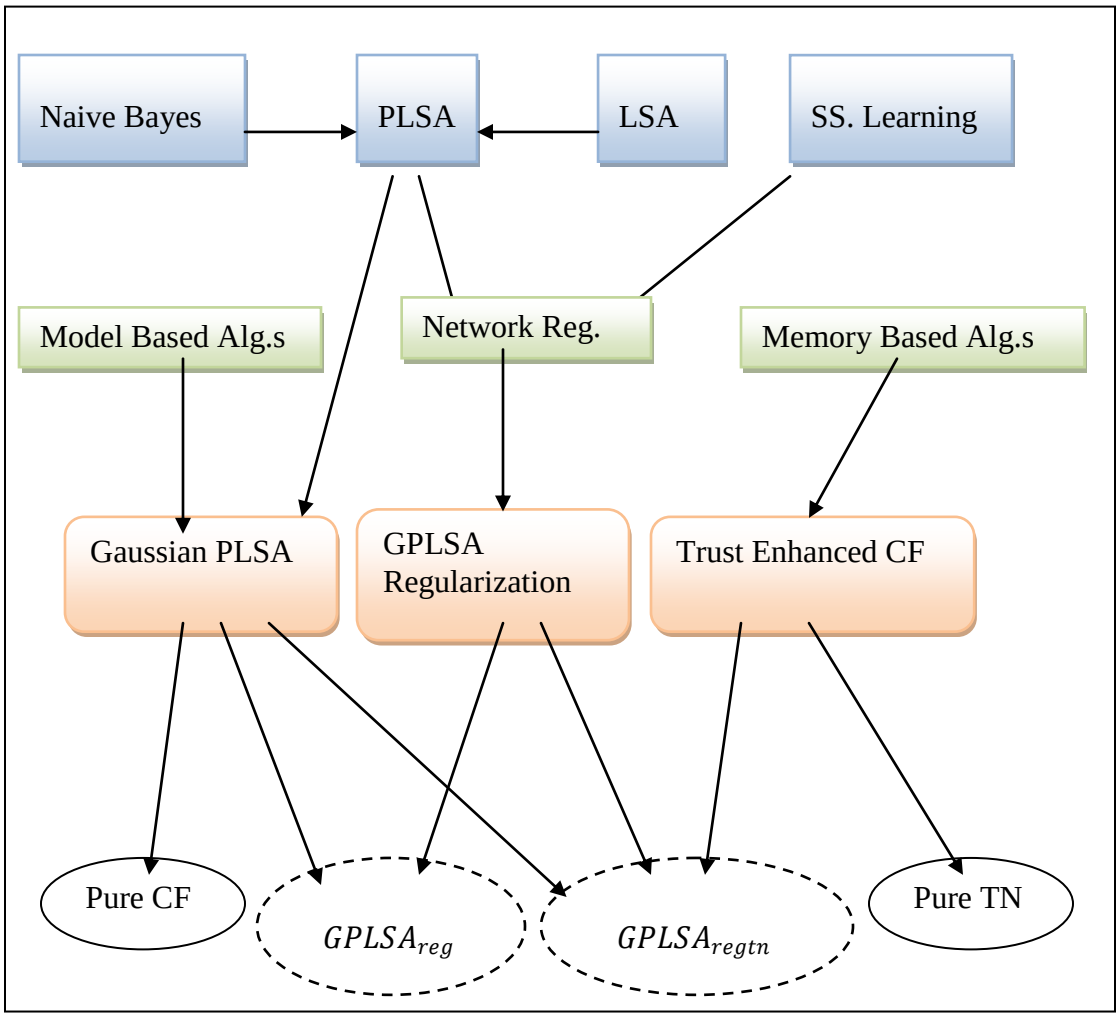


Figure 1 – Thesis Organization Overall Picture

Chapter 5 is dedicated to our experiments. The first experimentation set is the comparison of the model based approach, Gaussian PLSA and memory based approach. The second experimentation is the performance comparison of the trust enhanced recommendation algorithms. The last experiment includes the usage of indirect trust based methods as heuristics in the network regularization phase. The results of this experiment are compared with the sole CF algorithm, GPLSA and sole trust based methods.

Finally, in Chapter 6, we conclude the thesis and propose future directions of research based on this thesis.

CHAPTER 2

BACKGROUND INFORMATION

This chapter is dedicated to the background information. In this chapter, data terminology, background information on PLSA model and semi-supervised learning, which is the basis of network regularization approach, are reviewed.

2.1. Terminology About Data

There are two basic characteristics regarding the probabilistic model and the recommendation problem, namely dyadic data and bag of words assumption.

2.1.1. Dyadic Data

The text models that we mention in this chapter are actually applicable to any dyadic data. Dyadic data refers to being composed of two sets of discrete features, set A and B, such that the observations are triples (a,b,(observation|a,b)) where a ϵ A, b ϵ B and the observation value is conditioned on the value of a and b. In the domain of text modeling, the two sets are documents and words. One of the dyadic data models, the aspect model of probabilistic latent semantic analysis [1], has the symmetric interpretation where both documents and words are generated from hidden topics. Words and documents are conditionally independent from each other given the topics. Because of this generality, dyadic data models like probabilistic latent semantic analysis and Latent Dirichlet Allocation [75] can be used on any dyadic data. Based on this property, probabilistic latent semantic analysis has been applied on many domains. One of the domains is clustering/indexing of images where the observation dyads are image and local descriptor features [39]. Another domain is collaborative filtering. The

collaborative filtering domain aims to model the user behavior, and in this domain, the observation dyads are user-item pairs [55]. In bioinformatics domain, dyadic data appears in the form of gene expression for co-clustering/bi-clustering problems [84].

2.1.2. Bag of Words Assumption

Bag of words is the assumption that the order of the words in a document is negligible regarding its matrix representation. This assumption simplifies text models to reduce the computational cost. Under the bag of words assumption, to construct the matrix representation of a document, a word vocabulary is created that covers every distinct word as an index. If we are to represent documents inside a corpus, this vocabulary should be corpus-wide. A document is represented with an array of length |Vocabulary|, and the value of each word index on this array represents the number of occurrences of that word in the document. A corpus is composed of |Corpus| documents. Hence, the matrix representation of that corpus is a matrix of size |Corpus| x |Vocabulary|.

2.2. Background on PLSA

PLSA model is a probabilistic interpretation of the latent semantic analysis approach and an improved variant of the Naïve Bayes clustering algorithm. These two widely-known approaches are explained before going into the details of the PLSA model.

2.2.1. Latent Semantic Analysis

Latent semantic analysis (LSA) is overviewed to give an idea about the relation of probabilistic latent semantic analysis (PLSA) model to LSA, how the research evolved to the PLSA model and as an example of the model based algorithm that stems from linear algebra. For detailed information, reader can refer to [69]. The drawbacks related to polysemy-synonymy and the scalability-computational time of the vector space model [78],[69] are overcome by the latent semantic indexing method. The aim is to replace a large set of words with a smaller number of variables that can better represent the semantic of the word set.

The method relies on the singular value decomposition of the word-document matrix. The main steps of the latent semantic indexing are given below.

1- The rank of the word-document matrix is reduced by a singular value decomposition. To achieve this, k highest singular values are set to 0 which results in a k-dimensional representation of the original word-document matrix.
2- Perform the vector space methods on the dimension reduced matrices.

The singular value decomposition can be seen as choosing the dimensions of latent space in decreasing order so that the first dimension is situated at the direction with the largest deviation. However, there is no interpretation of the dimensions in the latent space and LSA still has the polysemy problem.

2.2.2. Naïve Bayes Model

Naïve bayes model [76] is the maximum likelihood approximation of the conditional probability distribution $P(C|F_1, \dots, F_n)$ where C is the class value and $F_{1..N}$ are the given features.

The Bayes formula is [76]:

$$P(C|F_1, .., F_n) = \frac{P(C)P(F_1, \dots, F_n|C)}{P(F_1, \dots, F_n)} \tag{2.1}$$

The denominator does not depend on the class variable *C*, so the joint probability is assumed to be sufficient to estimate the posterior.

$$\begin{aligned} P(C, F_1, .., F_n) &\sim P(C)P(F_1, \dots, F_n|C) \\ &\sim P(C)P(F_1|C)P(F_2, \dots, F_n|C, F_1) \\ &\sim P(C)P(F_1|C)P(F_2|C)P(F_3, \dots, F_n|C, F_1, F_2) \\ &\sim P(C)P(F_1|C)P(F_2|C) \dots P(F_n|C, F_1, \dots, F_{n-1}) \end{aligned} \tag{2.2}$$

With the simplifying assumption that features are conditonally independent, the following equation becomes valid.

$$P(F_i|C, F_j) = P(F_i|C) \tag{2.3}$$

Based on this assumption, the joint distribution can be expressed as follows.

$$P(C, F_1, \dots, F_n) = P(C)P(F_1|C)P(F_2|C) \dots P(F_n|C) \tag{2.4}$$

$$P(C, F_1, \dots, F_n) = P(C) \prod_{i=1}^{n} P(F_i|C) \tag{2.5}$$

$$P(C|F_1, \dots, F_n) = \frac{1}{Z} P(C) \prod_{i=1}^{n} P(F_i|C) \tag{2.6}$$

where Z is the normalization constant $P(F_1, \dots F_n)$. It is clear that a feature set can only belong to one class under naïve bayes model. In the text classification domain, this results in the assignment of one cluster value to each document. On the other hand, it is intuitive that actually documents belong to many clusters at the same time with a probability value under different contexts/viewpoints. The probabilistic latent semantic indexing/analysis proposes a model where different viewpoints can be estimated under solid statistical foundation.

2.3. Probabilistic Latent Semantic Analysis

Probabilistic latent semantic analysis[1] is an unsupervised clustering method. The method exploits the bag of words assumption for modeling documents.

For modeling dyadic data, PLSA proposes a probabilistic model. The model converges to real data via Kullback-Leibler divergence [77] through expectation-maximization iterations. The proposed model can be seen in Figure 2.

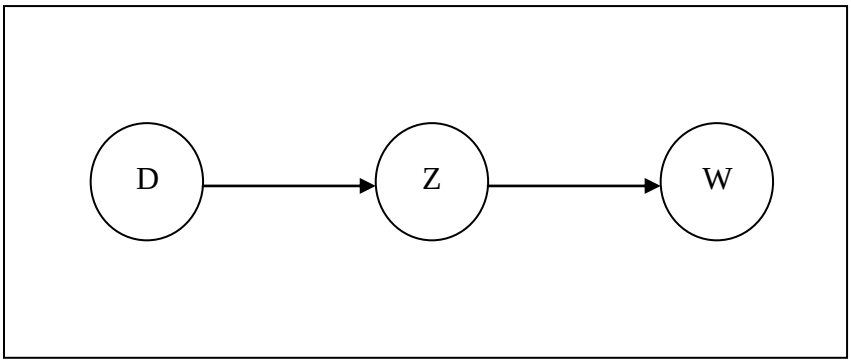


Figure 2 – Asymmetrical aspect model

This model shows that a hidden variable Z is used to map document variables to word variables which are observed. This is the asymmetric aspect model which is formulated by equation (2.8) below. The first formulation is an asymmetric model and the second model is symmetric around the hidden topic values which is given in Figure 3.

$$P(w,d) = \sum_Z P(z|d)P(w|z)P(d) \tag{2.7}$$

$$P(w,d) = \sum_Z P(z)P(d|z)P(w|z) \tag{2.8}$$

The two models are equal since $P(z|d) = \frac{P(d|z)P(z)}{P(d)}$.

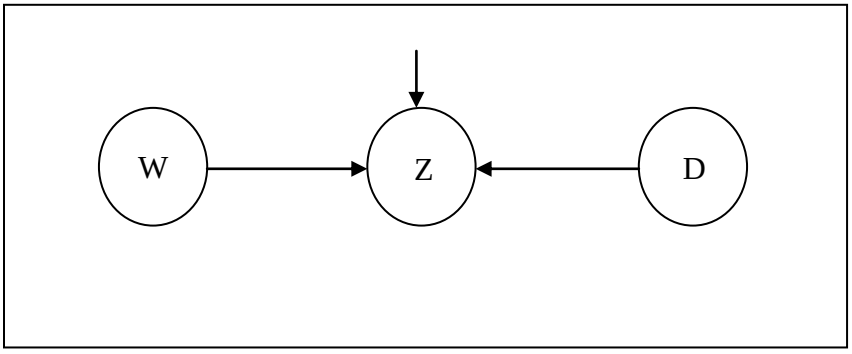


Figure 3 – Symmetrical aspect model

So, the free parameters are P(z|d), P(w|z) and P(z). Our aim is to maximize the likelihood of the model parameters given the data.

The likelihood function of the aspect model is given below [1].

$$L = \prod_{i=1}^{M} \prod_{j=1}^{N} P(w_i|d_j)^{n(w_i,d_j)} \tag{2.9}$$

where $n(d_i, w_j)$ represents the cooccurence of document *i* with word *j*.

In the case of PLSA, the free parameters of the model are P(z), P(d|z) and P(w|z). The posterior distribution is P(w,d,z) where hidden topics are marginalized out to get P(w,d). P(w,d) is the joint probability of word and documents which we aim to estimate.

The expectation maximization (EM) algorithm starts from a random position on the parameter space and tries to reach the optimum parameters using the log-likelihood as the control criterion. Once the change of log-likelihood on two successive iterations becomes less than a threshold value, the iterations are stopped.

Intuitively, the EM iterations try to fit the hidden topic marginalized posterior distribution to real data. The mentioned real data is kept in matrix $X_{d,w}$. PLSA can be seen as a way of factorizing the real data on three matrices P(z), P(d|z) and P(w|z).

The expectation maximization algorithm steps are summarized below.

Initial state:

X is a matrix of size |D| x |W| which is the observed document-word matrix

P(z) is a vector of length |Z| and it has a uniform initial value.

P(w|z) is a matrix of size |W| x |Z| and it has a normalized random initial value.

P(d|z) is a matrix of size |D| x |Z| and it has a normalized random initial value.

P(d,w,z) is a matrix of size |D| x |W| x |Z| and it has a normalized random initial value.

The expectation step is illustrated in Figure 4.

$$\mathrm{P(z|d,w)} = \frac{P(w|z)P(d|z)P(z)}{\sum_{z'} P(w|z')P(d|z')P(z')} \tag{2.10}$$

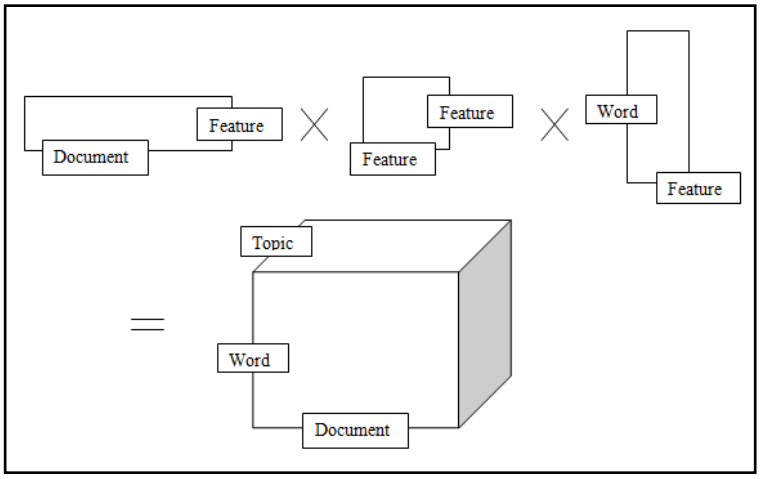


Figure 4 – Expectation step of PLSA

The maximization step is illustrated in Figure 5.

$$\mathrm{P(w|z)} = \sum_{D} X(d,w).* P(z|w,d) \tag{2.11}$$

$$\mathrm{P(d|z)} = \sum_{W} X(d,w).* P(z|w,d) \tag{2.12}$$

$$\mathrm{P(z)} = \sum_{W}\sum_{D} X(d,w).* P(z|w,d) \tag{2.13}$$

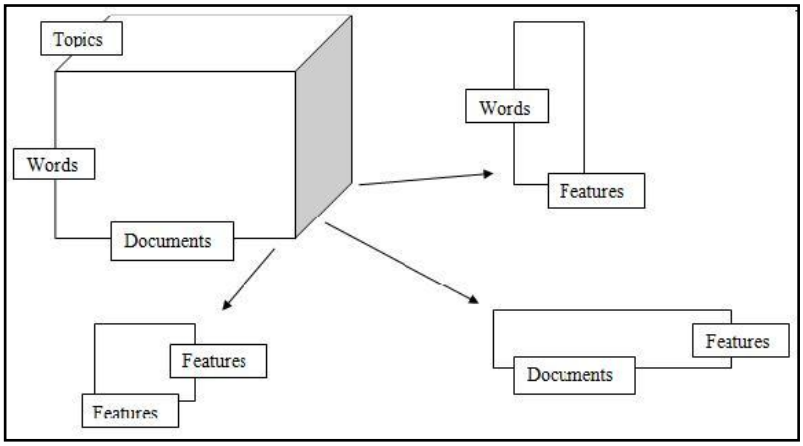


Figure 5 – Maximization step of PLSA

The log-likelihood function :

$$L = \sum_{Z}\sum_{D}\sum_{W} X(d,w).* \log(P(w|z)\ P(z)\ P(d|z)\ ) \tag{2.14}$$

The expectation maximization algorithm halts when the change in two successive log-likelihood values is below a threshold.

P(w|z) represents the probability that word w is generated from a hidden topic/cluster z. Each topic is represented as a word multinomial. This interpretation assigns a semantic to each cluster that is defined by word probabilities.

P(z|d) represents topic/cluster probabilities assigned to documents. P(z|d) can be found by P(z|d) $\propto$ P(d|z) P(z). Exploiting P(w|z) and P(z|d) together, one can reach the semantic meaning of each cluster, which, in turn, represents the captured semantic of the documents.

Topic models working under text modeling domain reflects the top-n words related to each document based on the above interpretation. The top-n words are the n words that have the highest probabilities given they belong to the kth topic. The highest topic probability given the document is found to define the top n words related to a document. The topic index is then assigned as the cluster/topic that document belongs to. Once the

topic of the document is assigned, the n words defining the topic are also assigned to the document.

2.4. Semisupervised Learning

The input data for a learning process can either be labeled or unlabeled. The labeled data includes feature values and a class information for each sample. The unlabeled data only includes feature values [49].

Classification is the supervised learning of the classes for the data samples. The data samples are split into training and test data. The samples of the training data are used with the class information provided. A classifier function is trained with this data. The classifier gives a class output for each test data sample. To test the performance of a classifier, the remaining test data samples are given to the classifier without the real classes information. This set of predicted class values are compared to the real class values of the test data. Finally, the classifier parameters are further optimized according to the classifier performance, if needed. Once the classifier is trained, new unlabeled data samples are classified using the trained classifier.

Clustering is the unsupervised learning of the classes. This approach is applied when we do not have class labels for the data to supervise the learning process. Clustering assigns class values to data points depending on the features of data. The performance of the clustering function depends on a defined error criterion. When the clustering function outputs class values for data, the error criterion is used to stop or continue the clustering process. The process is halted when the error criterion is below a defined threshold. Finally the trained clustering function is used to assign cluster values to new data samples.

In real life, labeled data is rare while unlabeled data is available in massive amount. Also, acquiring labels for unlabeled data via experts of the domain is an expensive and time consuming operation. But the high accuracy of training using labeled data is valuable. On the other hand, data is gathered without class information in its nature. Without labels, the learning accuracy is lower. So, a means to use high amount of

unlabeled data with comparably less labeled data needs to be devised. This approach is adressed in the semi-supervised learning research area [49].

Semi-supervised learning is the research area that tries to enhance clustering and classification procedures by using labeled and unlabeled data together in the same learning process. This approach aims to reduce the expensive data labelling effort with the usage of easy-to-find unlabeled data. It is most of the time impossible to define a class value to data beforehand, although the high price and long time of labelling is accepted. The alternative of giving exact labels to data is defining link probabilities of data. The link probability between two samples can be defined as the probability that the two samples reside in the same cluster. Again, it is hard to obtain probability values for the linkage of two samples. Instead, to ease the data labelling process, the linkage information is obtained in binary. This is referred as pairwise constraints, must-link and cannot-link data pairs [47,48,49]. The error criterion of clustering process favors two must-link samples residing in the same cluster and two cannot-link samples residing in different clusters. Likewise, the violation of the link information results in penalization of the learning process. The K-means algorithm can be viewed as a variation of the expectation maximization algorithm. The number of clusters is taken as input to the algorithm which represents the number of mean cluster values for the k clusters. The basic k-means algorithm is given below. The expectation and maximization steps are iterated successively until all data samples are processed.

2.4.1. K-Means

- Initial Step:
  - Randomly select k samples and take the sample values as the mean values of k clusters.
- Expectation(Assignment) Step:
  - Take a new sample and assign it to the cluster which has the closest mean value to the new sample.
- Maximization(Update) Step:
  - Calculate the new k cluster mean values

$$Objective_{kmeans} = \sum_{x_i \in X} \left\| x_i - \mu_{l_i} \right\|^2 \quad (2.15)$$

The basic semi-supervised approach is employed on k-means algorithm in [2] and it is shown that semi-supervised learning significantly improves the unsupervised k-means with the usage of pairwise constraints.

Another variation of semi-supervised k-means algorithm is called seeded k-means. The main idea is to use the semi-supervision in the initial step. The semi-supervision can be in the form of cluster values for each sample or in the form of pairwise constraints. When the semi-supervision is given as cluster values, it is trivial to firstly choose the labeled data as the cluster centers and then assign the unlabeled data to these clusters [50].

The following algorithm shows how semi-supervision can be applied to the updating step. The penalization of violated pairwise constraints is added to the k-means algorithm as defined below.

2.4.2. Cop K-Means

- Initial Step:
  - Randomly select k samples and take the sample values as the mean values of k clusters.

- Expectation(assignment) Step:
  - Take a new sample and assign it to the cluster which has the closest mean value to the new sample. The new assignment should not violate any cannot link constraints. (i.e the new cluster shouldn't contain any samples which have a cannot link constraint to the new sample.) If the assignment violates a constraint, assign the new sample to the next closest cluster, until no constraints are violated. If no clusters are available, the algorithm fails to assign the sample to a cluster.

- Maximization(update) Step:
  - Calculate the new k cluster mean values.

This algorithm can be simply viewed as minimizing the following objective function.

$$Objective_{pckmeans} = \sum_{x_i \in X} \left\| x_i - \mu_{l_i} \right\|^2 + \sum_{(x_i, x_j) \in M} w_{ij}\, \ell[l_i \neq l_j] + \sum_{(x_i, x_j) \in M} \bar{w}_{ij}\, \ell[l_i \neq l_j] \tag{2.16}$$

Here, l_i denotes the cluster of sample x_i and ℓ is an indicator function where ℓ[true]=1 and ℓ[false]=0.

Another improvement on k-means is achieved via metric learning [11]. The objective function of k-means measures the Euclidean distance of a sample to the k cluster means. The constrained k-means adds pairwise constraint violations. When a sample assignment is not satisfiable according to the constraints, the constrained k-means fails to assign the sample to a cluster. This disadvantage is overcome by introducing weights to violations which transforms an unsatisfiable state into a low satisfied state. At the same time, the weighting scheme should take into consideration that two nearby samples having a cannot link constraint must be penalized more than two distant samples. Likewise, two nearby must link constrained samples should be penalized more than two distant samples. This idea is encoded in the below metric learning objective function.

$$Objective_{metrickmeans} = \sum_{x_i \in X} \left(\left\| x_i - \mu_{l_i} \right\|_{A_{l_i}}^2 - \log\left(det\left(A_{l_i}\right)\right) \right) + \sum_{(x_i, x_j) \in M} w_{ij}\, f_M\left(x_i, x_j\right) \ell\left[l_i \neq l_j\right] + \sum_{(x_i, x_j) \in M} \bar{w}_{ij}\, f_C(x_i, x_j) \ell[l_i \neq l_j] \tag{2.17}$$

Here, f_C is a penalty function for the violation of a cannot link between two sample points x_i and x_j that takes into consideration the distance between x_i and x_j . Similarly, f_M is a penalty function for the violation of a must link between two sample points. W and $\bar{W}$ are constraint costs. A_{l_i} represents cluster specific metric weights. Without going into further details, the metric pairwise constrained k-means algorithm iterates through cluster assignment, mean estimation and metric update steps. The cluster assignment is obtained by choosing the cluster for the new sample that minimizes the above objective function. The mean estimation is solely the updated cluster mean calculation. The cluster specific metric weight matrix A is updated only for the cluster that the new sample is assigned to. Further details can be found in [12].

CHAPTER 3

RELATED WORK

In this chapter, we first give an overview of recommender systems. The problem of recommendation is stated and the methods in this thesis are categorized. Also, the details and the terminology of the data are explained.

We give a detailed review of the articles in the literature that deal with integrating constraints into the PLSA model as semi-supervision.

Another dimension of our work is the probabilistic models of recommendation. We give a detailed survey on recommendation models based on PLSA. These models are explained in detail to provide a complete view of state of the art.

As mentioned in the previous chapter, PLSA based models have the ability to be implemented on a distributed setting. The corresponding methods regarding the scalability of PLSA model is explained in this chapter.

3.1. Recommender Systems Overview

The recommendation problem can be seen as approximating empty cells of a huge two-dimensional matrix. In this matrix, each cell is a rating value. Each row of the matrix corresponds to a user's ratings to all items and each column of the matrix corresponds to an item's ratings given by all users. The item can be any entity of interest, e.g. a movie, a product, a document, or a news article.

Recommender systems are generally categorized as content-based, collaborative and hybrid methods.

3.1.1. Collaborative Filtering

Collaborative filtering methods define the user interests by the rating patterns of the user on the item usage history. On the other hand, content-based methods define the user interests based on the content of the items that the user has previously used.

Recent work on recommender systems have emphasized the usefulness of the collaborative data while the data content is seen less important in representing the user interests [81].

As pointed out in [79], there are several problems in collaborative filtering that are summarized in the following.

- *Cold start problem*: Collaborative filtering method needs to have a sufficient amount of declared ratings to successfully represent the user's interest.
- *Sparseness problem*: Rating patterns on collaborative filtering are based on the whole set of items and each user can declare rating on only a very small percentage of these items.
- *First rater problem*: If a new item is added to the system, this new item cannot be recommended by collaborative filtering methods until a user declares a rating about it.
- *Popularity bias*: Based on the first rater problem, the older and more popular items are favored over new items on collaborative filtering. This causes a problem about diversity of the items recommended and lack of representing unique tastes in the system.

3.1.2. Content Based Filtering

Content based filtering extracts features from items and tries to model users with these features and the rating values of items. Content based filtering, as the complement of collaborative filtering, faces following problems.

- *Insufficient content*: Item content may not be sufficient to represent the user interest.
- *Misleading content*: Item may contain inaccurate content.
- *Feature extraction*: An item content representation needs to be constructed as features. This phase may result in a suboptimal representation.
- *Ideas of others*: Judgments of other users are ignored.

3.1.3. Hybrid Approaches on Recommender Systems

A hybrid recommender system combines different recommendation approaches. The combination of collaborative filtering and content based filtering has been the main hybridization approach in recommender systems. Other hybridization approaches are also applied, i.e. combination of CF and social networks, trends.

Collaborative and content based methods have advantages and disadvantages over each other. The collaborative methods rely on rating patterns. In order to significantly represent a user, the user should have provided a sufficient number of declared ratings. So, this approach provides a more accurate interest representation by each additionally declared rating. However, the content-based interest representation relies on the content of the items that have been rated and the content is available initially throughout the system's lifetime. Hybrid recommender systems which combine content based and collaborative methods are proposed to deal with these problems.

According to the survey of Tuzhilin et. al. [80], different hybridization techniques are grouped as follows.

- Combination of collaborative filtering and content based filtering in the rating prediction phase
- Projecting content based characteristics to collaborative filtering
- Projecting collaborative filtering characteristics to content based filtering
- Unifying the content based and collaborative approach in a single algorithm

Besides the rating behavior and item content, it is also possible to integrate additional data sources like social networks [25, 26] and trends [3, 93], although these two data sources are not directly given as categories of recommender systems in the literature.

A social network provides a user to user similarity which collaborative filtering aims to capture. This aspect can be thought of as a user based content data in the form of other related users. This information provides another means to deal with the cold start problem. Besides, the user can freely favor other users that appeal to his/her unique interests which provides a solution to the popularity bias problem. Along with these benefits, it does not carry the problems of content based approach regarding the content feature extraction. But this aspect has its own challenges considering its integration to collaborative filtering, predicting missing links in the graph and trust propagation.

Along with the theoretical advances on recommender systems, commercial products have been also effective in leading the research direction. 2009 Nielsen commercial research [8] result given in Figure 6, shows that most of the users act according to their friends' behaviour. This aspect of the recommendation problem has been adressed in the social recommendation research community.

Trends are another beneficial data source effective in the rating prediction. Trends are local patterns that change with respect to time and place. Based on the available data, both of these aspects have been incorporated to rating prediction in the literature [79].

Another aspect of categorization on recommender systems is based on the solution methods. From this point of view, the recommender systems are categorized into memory-based and model based methods.

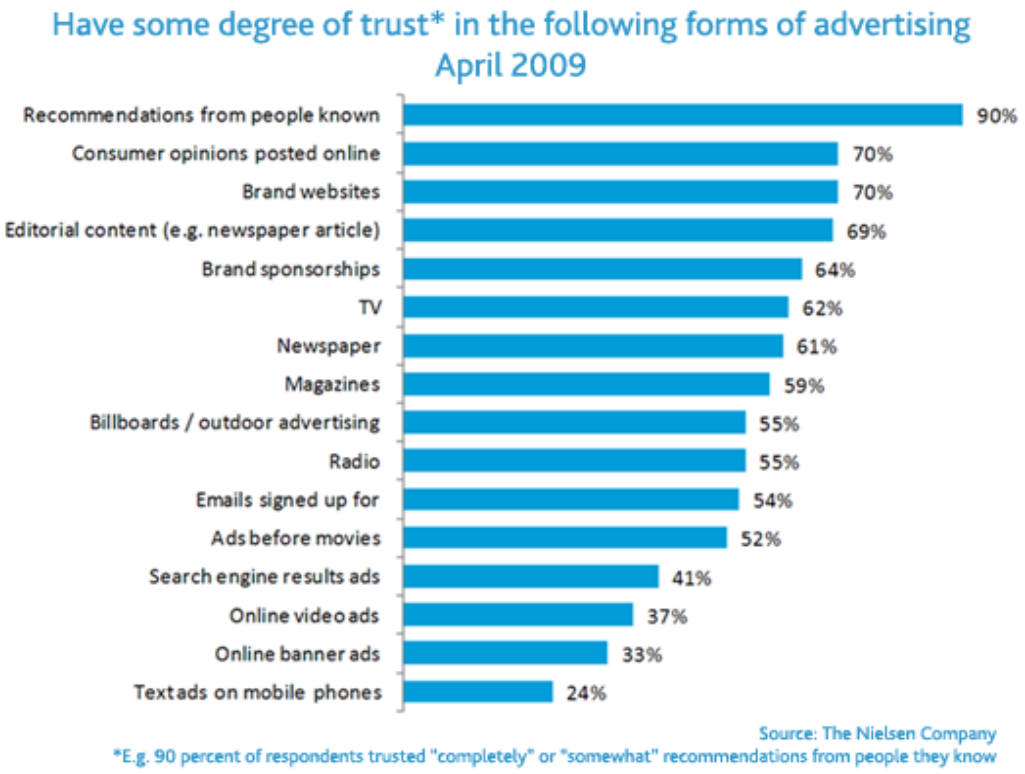


Figure 6 - Nielsen commercial research results [9]

3.2. Memory Based Recommendation Algorithms

Memory based methods operate over the whole input rating data online and they have a high complexity of rating prediction. Memory based algorithms, also referred to as heuristic algorithms, follow common operations. These algorithms mainly calculate user to user and item to item similarity over the whole user-item matrix with a similarity metric. It can be infeasible to use memory based methods online because of the high dimensionality and sparsity of the user-item matrix. Memory based methods are not based on lower dimensional representation of the input data and can be modified with heuristics to achieve a higher accuracy with the cost of complexity. The most common technique for rating prediction on memory based methods is the k-nearest neighbor based algorithms, generally referred to as top-n recommendation algorithms.

The existing approaches for the memory based methods use some common operations [85]. Algorithms under memory based category are based on assessing the similarity of

users/items and ranking according to the similarity level. The similarity information is obtained via similarity metrics. In the following, three most popular similarity metrics on memory based recommendation algorithms are given. Next, two basic memory based rating prediction algorithms are presented.

3.2.1. Similarity Metrics

Similarity metrics are methods of defining the distance between two vectors based on different prior information that change according to the problem. $r_{a,i}$ denotes the rating value that user *a* has given to item *i*. $\overline{r_a}$ is the mean of the rating values that user *a* has declared.

Cosine Similarity

One of the popular methods for relevance ranking in the vector space model [78] is the cosine similarity method. Given two vectors v_1 and v_2, cosine of the angle between the two vectors defines the similarity of vectors v_1 and v_2.

$$\text{Cosine similarity} = \frac{v_1 \,.\, v_2}{\|v_1\|\|v_2\|} \tag{3.1}$$

a . b is the dot product of vectors a, b and ||.|| is the length of the vector.

Pearson Correlation Coefficient

Pearson correlation coefficient [86] represents the similarity of an active user *a* to another user *b* according to their ratings.

$$Pearson - sim_{u,a} = \frac{\sum_{i=1}^{m}\left(v_{a,i} - \overline{v_a}\right)\left(v_{u,i} - \overline{v_u}\right)}{\sqrt{\sum_{i=1}^{m}\left(v_{a,i} - \bar{v}_a\right)^2 \sum_{i=1}^{m}\left(v_{u,i} - \bar{v}_u\right)^2}} \tag{3.2}$$

Adjusted Cosine Similarity

Adjusted cosine similarity is used to measure the similarity of items. This method also considers the varying rating scales of users.

$$Adj.Cosine\ sim_{i,j} = \frac{\sum_{u\epsilon U}\left(v_{u,i} - \overline{r_u}\right)\left(v_{u,j} - \overline{v_u}\right)}{\sqrt{\sum_{u\epsilon U}\left(v_{u,i} - \bar{v}_u\right)^2}\sqrt{\sum_{u\epsilon U}\left(v_{u,j} - \overline{v_u}\right)^2}} \quad (3.3)$$

3.2.2. Top-N Recommendation

Top-n recommendation is a way of finding out n most related items or users and predicting the unobserved rating based on the n items or users. The user and item based approaches are given in this part.

An example for item based recommendation is given in Figure 7. Let us assume the active user has already rated $I_1$and I_4. In the first step of item based top-n recommendation, similarity between items are calculated. For each item, k-most similar items are kept. In our example in Figure 7, k is 3. So, 3 most similar items for each item are kept and other items are removed. The removed similarities are shown as strikethrough similarity values. On each item column, we add the similarities with the active user's items. We finally remove the items that the active user has already rated and take the N items that have the highest scores, shown with an additional strikethrough on similarity values in Figure 7. The item ranking for the active user, given the rating matrix in Figure 7 is I_5, I_2 and I_3.

In the first step of user based top-n recommendation, we need to find k nearest user neighbours to do recommendation based on similar user preferences. In the case of our demonstration, given in Figure 8, assume k=2 and two most similar users to the active user are U_3 and U_4. To predict the unobserved items for the active user, the algorithm averages over the ratings of the k neighbour users' profile. The unobserved item I_5 is given rating 2, based on the average votes of the two neighbours. There can also be a weighting scheme that would replace the averaging on this algorithm to further enhance the recommendation performance.

	I_1	I_2	I_3	I_4	I_5
I_1	~~0.2~~	~~0.1~~	0.3	~~0.4~~	~~0.1~~
I_2	~~0.8~~	~~0.3~~	~~0.1~~	~~0.6~~	0.9
I_3	~~0.4~~	0.5	~~0.2~~	~~0.8~~	0.7
I_4	~~0.2~~	0.5	0.6	~~0.2~~	~~0.1~~
I_5	~~0.6~~	~~0.1~~	0.4	~~0.3~~	0.5
	~~0.45~~	0.5	0.5	~~0.6~~	0.7

Figure 7 – Item based top-N recommendation

	I_1	I_2	I_3	I_4	I_5
U_1	3	2	4	1	0
U_2	3	2	4	5	1
U_3	1	3	4	3	3
U_4	1	3	3	2	1
U_a	1	3	4	3	0
	1	3	3.5	2.5	2

Figure 8 - User based top-N recommendation

3.3. Model Based Recommendation Algorithms

The model based algorithms construct a lower dimensional representation of data. The number of parameters in the resulting model is kept much less than the unknown parameters of the problem. Model based algorithms stem from two different backgrounds, probabilistic graphical models and linear algebra. Models that stem from linear algebra adopt singular value decomposition [69] for low dimensional representation. Singular value decomposition removes redundant, noisy and unrepresentative users/items from real data to reduce data sparsity. Methods that incorporate singular value decomposition are [3, 65]. The most recent research on matrix factorization [65] via optimization can be traced back to singular value decomposition method. On the other hand, probabilistic graphical models are based on solid statistical foundation. The probabilistic models provide interpretability of methods under a common mathematical framework. The equivalence of many matrix factorization methods and their probabilistic counterparts is proven in the literature [24]. Probabilistic latent semantic analysis, which is the probabilistic interpretation of latent semantic analysis, improves the naïve Bayes model [76] with its ability to assign a probabilistic membership to every cluster. The improvement of probabilistic latent semantic analysis over latent semantic analysis is its ability to represent polysemy. The survey of probabilistic models in 3.5 includes many model based approaches [1, 20, 21, 62, 63, 68].

3.4. Probabilistic Models for Recommendation

This section explains the probabilistic recommendation models in the literature. The diversity of researchers that come from different backgrounds leads to the need to unify the proposed methods under some mathematical framework. Probabilistic graphical models have been proposed as the base for the mathematical framework of recommender system methods in [37]. These methods provide a complete means for processing data, creating statistical models specific to the problem and fitting the model using the available partly structured data, according to which the model is constructed to reach the solution. Topic models [74], which is a field working on document

clustering and text processing, also use probabilistic graphical models as a basis. Methods under topic models try to reach the semantic meaning of documents and words. Works of researchers coming from this field attracted significant attention recently. This idea, with no loss of generality, is applicable to the problem that recommender systems deal with. A recent article with this background is awarded as a distinguished work and won best paper award at RecSys '09 [83].

Probabilistic models have been used both to accurately learn rating patterns and to combine several data sources in a principled way, which is referred to as multi-way models.

3.4.1. Rating Learning Models

Rating learning models follow different probabilistic methods or statistical models as the collaborative filtering approach. In this subsection, various probabilistic rating modelling approaches are reviewed. The work in [9] proposes a model to capture the personality types and model the ratings based on the personality types. The first assumption of this model is that the observed ratings of a user has a Gaussian noise with mean equal to true ratings and standard deviance, σ, is the free parameter.

$$P(V_{ij} \mid V'_{ij}) \sim e^{-(V_{ij} - V'_{ij})/2\sigma^2} \tag{3.4}$$

Depending on the context, the user's mood and trends, the user may give different ratings to the same movie. Furthermore, when the user's personality type is known, the ratings of the user are assumed to be independent. Another assumption is that the user rating vectors are representative of the underlying personality types. The most probable rating value is returned. The structure of the personality diagnosis is the same as the naive Bayesian model as shown in Figure 9. Personality diagnosis model is shown to outperform vector similarity and simple correlation based models.

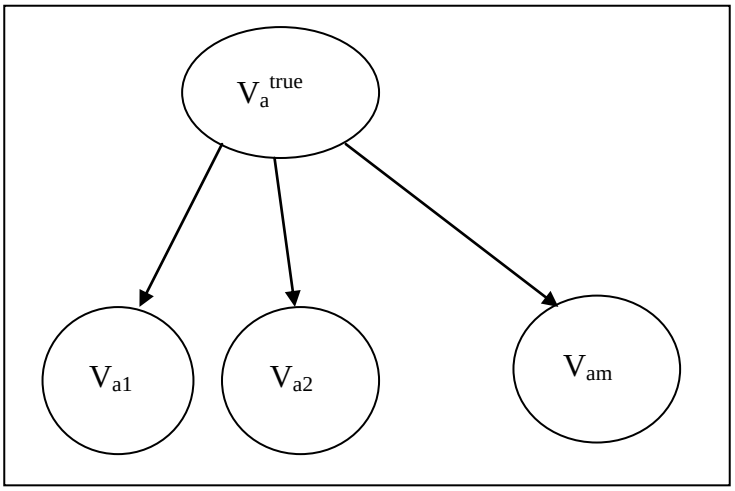


Figure 9 – Personality diagnosis model

The term flexibility, is defined as letting users and items to be members of more than one class. This is achieved by introducing latent variables between both user-rating observations and item-rating observations. The flexible mixture model [20] also aims to predict the variance in the user rating behaviour with the normal distribution while the mean of the normal being the observed true rating. This is done by introducing two latent variables to account for variability and user interest given the user and the rating. The graphical model for the flexible mixture model is shown in Figure 10. Since the ultimate goal of collaborative filtering is the prediction of ratings for a specific user, the fold-in approach [1] is used. The idea is to approximate the joint probability distribution, $P(u, y^t, v)$, where y^t is the test user. The rating expectation for that user of an item y can be computed as below.

$$\hat{V}_{y^t}(u) = \sum_{v} r \frac{P(u, y^t, v)}{\sum_{v'} P(u, y^t, v')} \tag{3.5}$$

The work [20] also mentions the varying rating behaviours of users since a user can rate his/her most disliked movie 3 while another can rate it 1 and similarly a user can give 5 while another can give 4 as the rating. This is handled with the user normalization step in [68]. A second improvement of the flexible mixture model on Gaussian PLSA is the integrated calculation of user normalization and also the calculation of item normalization via hidden variables.

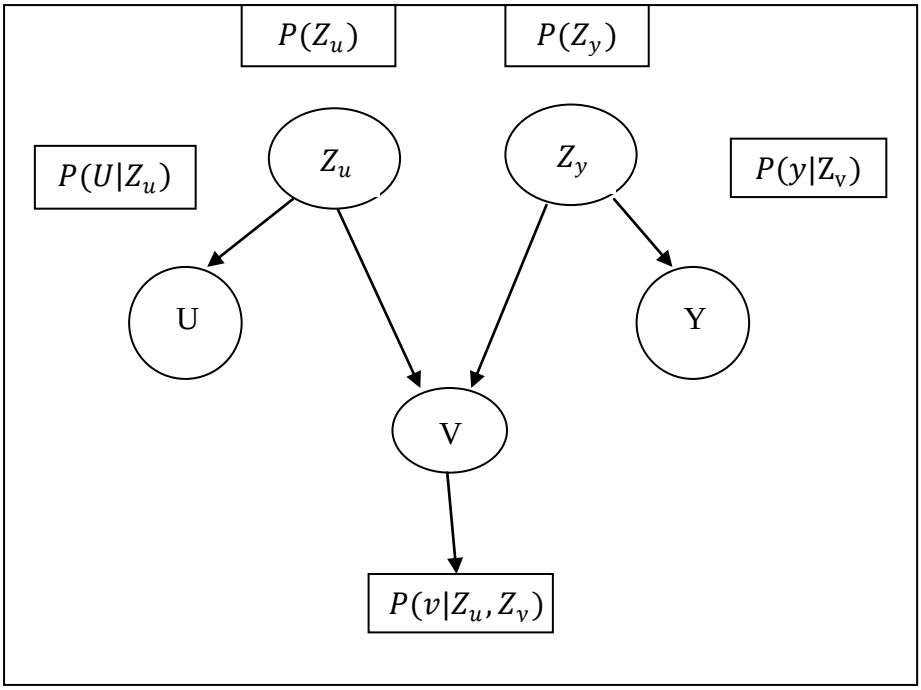


Figure 10 - Flexible mixture model [20]

This work also shows how the rating values can be modelled as a continuous random variable. In this model, the variable V, called the preference node, accounts for the true preference values and Z_v accounts for different rating behaviours. The proposed model can be seen in Figure 11. The generation of an item Y is jointly affected by latent item variables Z_y and latent rating variables, Z_v. In this setting, users are clustered according to both user interest patterns and rating behaviors. The generation of the preference node V is jointly affected by nodes Z_u and Z_y. The generation procedure of a rating takes preference node and Z_r into consideration. On the model, user and item latent variables are not decoupled over rating variable to separate preferences and rating behaviors while the rating behaviors can still be affected over the preference node.

The V and Z_r models make inference and prediction computationally very expensive. This vulnerability is maintained by relaxing the model to a simpler one. The aim is to use the simple model to acquire the preference value directly and using a modification of flexible mixture model to find ratings for the test user given the test set. The modification of the flexible mixture model only comprises replacement of node R with preference node V.

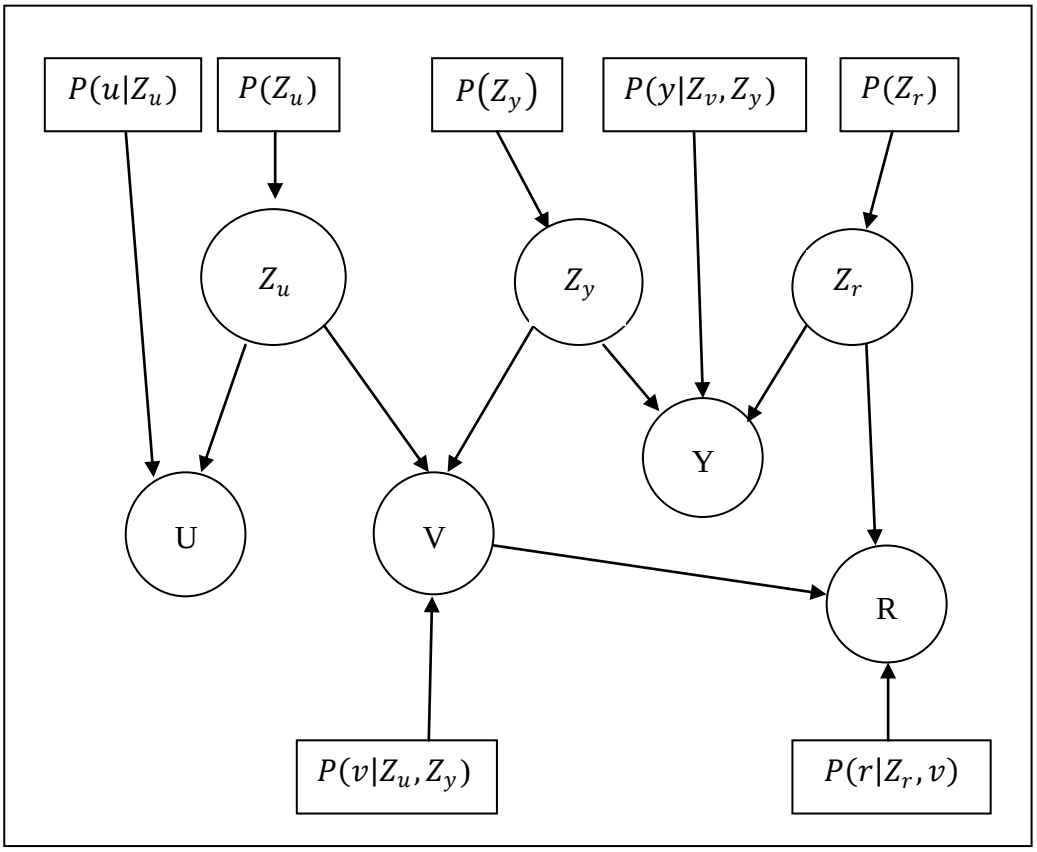


Figure 11 - Flexible mixture model with user normalization

The simple model is referred to as the decoupled model. The decoupled model models the actual preference value of an item rated 'r' for a user given a set of rated items from the user. Two factors are said to affect this model. Firstly, the ratio of items rated less than 'r' is effective. Secondly, ratio of items rated 'r' is effective on the model. For the experimentation, personality diagnosis, aspect model and a person-correlation coefficient based method are implemented. It is shown that flexible mixture model outperforms others based on the MAE performance criterion. Active learning method can fill the inadequacy that is the cold start problem of collaborative filtering method. Supposing a new user has entered the system, the user is queried by the system with movies to be rated by him/her. The selection of movies is random at default. But it is intuitive that some method can be used instead of the random choice. The work in [63] compares random and Bayesian movie selections as active learning approaches and shows the system's curve of learning the user's profile over the increasing number of movie choices.

3.4.2. Multi-way Models

This subsection reviews the related work on utilizing various data sources for data modeling. On probabilistic modeling domain, this approach is referred to as multi-way models. A multi-way model that models users via documents and words, given in Figure 12 is proposed in [25, 30] on document classification domain, which is the basis of the multi-way models on the recommendation domain.

The authors of [19] introduce two-way and three-way models showing how mixed events affect the same semantic space. Two-way models are actually the probabilistic latent semantic analysis approach applied on different two-mode co-occurence data. All steps of the probabilistic latent semantic analysis apply for the given two-way models in the paper. Two-way model data are user-item, user-actor and user-director co-occurrence values.

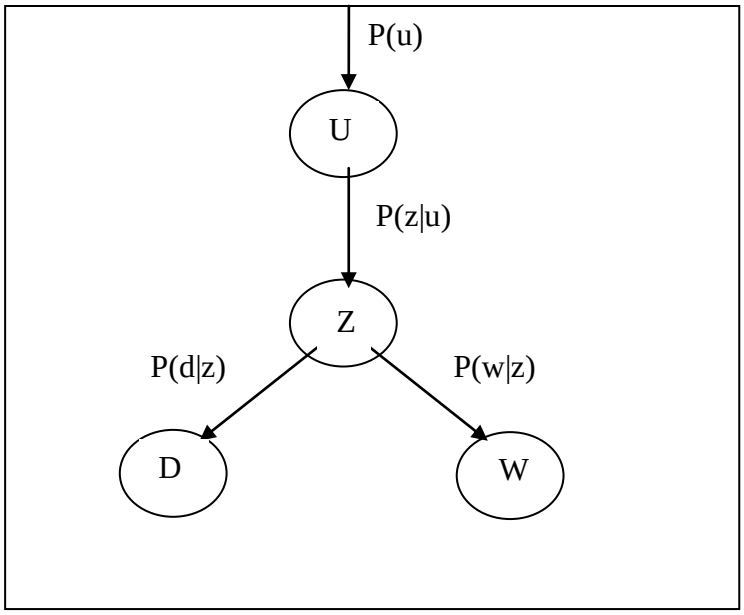


Figure 12 - Probabilistic hybrid model

To project new items on the latent semantic space created using user-actor or user-director models, the fold-in approach is followed which is proposed in [1]. This approach is used to project term-query in word-document semantic space to find the relevance of the new query to the documents that are already available in the corpus. The usage of folding-in is explained in the movie domain. The fold-in for the user-item

model is the same as document-word model that has been proposed in the work of Hoffmann [1]. But folding-in of a new user-item instance also requires to fold-in new user-actor and new user-director instance to the corresponding models. The change on topics caused by the actors of the new movie is transferred to the item semantic space. Once the item semantic space is updated, it is trivial to find user probability given the item using the available user semantic space on user-item model.

Another approach mentioned in the paper is the three-way aspect model. This model unifies two of the two-way models in a similar manner used on two-way models. The unified model assumes the three-way data independent given the topic. Two three-way models are applied on the work. First model incorporates user-movie-actor data and the second model incorporates user-movie-director data. Since both models are same, only the first model's expectation maximization steps are explained.

On the generative process, firstly the topic is chosen. Then user, movie and actors are chosen given the topic. The last approach mentioned on this work is call mixed event-space models, where different model parameters are combined.

The mixed event space experiments are mixing two two-way event spaces and mixing three two-way event spaces. The authors compare the two mixed event approaches and propose the hypothesis that the three two-way mixed event models are unnecessarily complex.

3.5. Pairwise Constraint Supervision into PLSA Model

The main aim of this part is to present the approach that inspired the idea of using social networks in the collaborative filtering process, which is the basis of this thesis. This approach assumes that there is either a must-link or a cannot-link pairwise constraint between each document in the corpus. In other words, the constraints are global according to this method. To use social networks, we look for pairwise constraint integration but we need the constraints to be local. The methods [29], that we present based on our domain, can handle similarity networks based on local consistency. For

completeness of the thesis and as the inspiration for our application, we explain pairwise constraint semi-supervision method in this part.

This approach proposes a method to incorporate pairwise relations between documents. Authors of [5] work on the classification of text data from different domains that have common concepts. Text documents from different domains are given with a simple categorical information that defines the relation between top-category concepts and sub-category text corpora. Along with the text data annotated categorically, unlabeled text corpora is available. The aim of the authors is to improve clustering performance using both categorically annotated and unlabeled text corpora. They also expand the work to employ semi-supervised learning approach on probabilistic latent semantic analysis.

The authors exploit the common word-topic matrix for different domain annotated text data and unlabeled text data. This can be thought of as transferring the word relations from one domain to another. The value for number of topics is kept same for all domains and the unlabeled data. A common vocabulary is used in the bag of words representation of each corpus so that the document representations are given with the same word-topic distributions, $P(w|z)$. It is shown that instead of training two probabilistic latent semantic analysis models for labeled and unlabeled data, one model can be used to integrate the two models jointly.

Moreover, the authors integrate semi-supervised learning to the PLSA model. The semisupervision is applied in the form of pairwise constraint information on the labeled documents. The pairwise constraints are available as must-link and cannot-link constraints. The integration of the semi-supervised approach is achieved by introducing a weight parameter. In Figure 13, d_u denotes unlabeled documents and d_l denotes labeled documents.

The must-link pairwise constraint is applied on the log-likelihood of the model by a weight parameter that affects the probability that two documents are generated from the same topic.

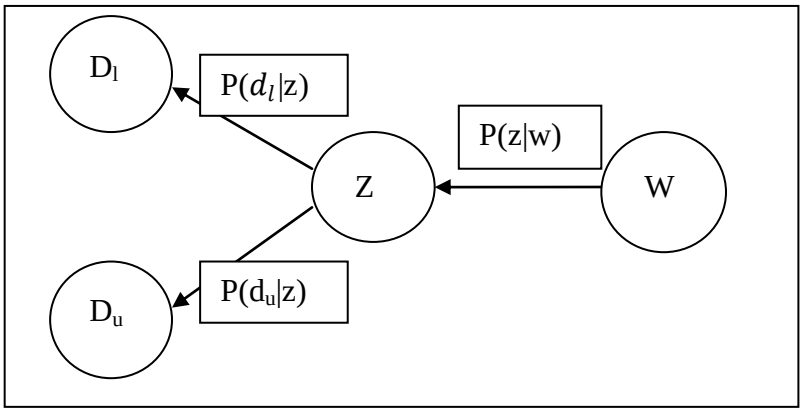


Figure 13 – Semisupervised document clustering model

Having the complete log-likelihood violated by the pairwise constraints, helps decide when to stop the expectation maximization iterations. But a way that considers the pairwise constraints to modify the document-topic subspace is also needed. The complete log-likelihood of the data and the objective function, L_c , is non-convex and reaching a globally optimal solution is hard to obtain. [6] The two constraints, added to the log-likelihood of the standard probabilistic latent semantic analysis, avoids the use of expectation maximization. The proposed method to optimize the new objective function L_c is minorize-maximization for its capability to optimize complex objective functions and various usages in machine learning.

Minorize-maximization tries to calculate a series of easier optimizations instead of maximizing a difficult one. This easier function is called the minorizer which should satisfy the below constraints.

$$L_c(\theta) \geq M\left(\theta, \theta^{(t)}\right)$$
$$L_c\left(\theta^{(t)}\right) = M(\theta^{(t)}, \theta^{(t)}) \tag{3.6}$$

$\theta^{(t)}$ denotes the last values of parameters during the minorize-maximization steps. The parameter values at the next iteration, $\theta^{(t+1)}$. Assuming $M\left(\theta, \theta^{(t)}\right)$ to be the minorizer of the complete log-likelihood of the model, below statements hold.

$$L_c\left(\theta^{(t)}\right) = M\left(\theta^{(t)}, \theta^{(t)}\right) \tag{3.7}$$

$$L_c\left(\theta^{(t+1)}\right) \geq M\left(\theta^{(t+1)}, \theta^{(t)}\right) \geq M\left(\theta^{(t)}, \theta^{(t)}\right) \tag{3.8}$$

$$L_c\left(\theta^{(t+1)}\right) \geq L_c\left(\theta^{(t)}\right) \tag{3.9}$$

This shows that the minorizer and complete log-lilkelihood functions are non-monotonically increasing.

Minorize-maximization algorithm guarantees that complete log-likelihood is increasing and converges to a local optimum.

Authors use the P(z|d) values as document features after convergence and use these features to cluster the documents.

On the experimentation part, the algorithm is shown to perform better than other tested algorithms, transductive support vector machine, support vector machine and naive bayes classification. The tests for model's sensitivity to parameters showed that the model is not very sensitive to pairwise constraint parameters, β_1 and β_2 .

Another approach on semisupervised PLSA is proposed in [67]. The work in [67] proposes a semi-supervised version of the probabilistic latent semantic analysis. The semi-supervision is achieved by data labeled directly with class information, not via pairwise constraints. The usage of probabilistic latent semantic analysis is based on the assumption that each hidden topic component represents the probability of inclusion to a corresponding cluster. So, the number of clusters equals to number of dimensions in the hidden variable. The fake label is introduced as the additional dimension to the hidden variable. Likewise, the given true label information is added to hidden variable as an additional dimension. The main idea of the authors is to employ labeled and unlabeled data together. This is achieved by assigning a fake initial label for all the unlabeled data and during the expectation maximization iterations all the labeled data will keep their labels whereas so called "fake" labels of the unlabeled data are expected to converge to their true value.

3.6. Scalability of PLSA Models

The authors of [66] apply the probabilistic latent semantic analysis approach on a text corpus for document clustering. The article mainly discusses the unnecessary storage of the posterior distribution in memory since the posterior distribution is neither an input nor an output of the algorithm. To emphasize this better, authors modify the definition of the algorithm as below.

$$Q(\text{topic})^{it} = \{ P(w|topic_1)^{it-1} * P(topic_1|d)^{it-1} \}^{\beta(it)} \quad (3.10)$$

Each Q variable is of size WxD and it denotes iteration.

$$Q_{sum} = \sum_{topi\ c_n \epsilon T} Q(topic_n) \quad (3.11)$$

Another property of the probabilistic latent semantic analysis algorithm discussed in the article is independence of calculations topic-wide. In one iteration, the normalization constant is calculated. Then, for each topic, the free parameters and the posteriors are calculated. Since these operations are independent once the normalization constant is known, the operations are said to be distributable among processors. Apart from the above properties, authors observed that the number of topics is generally much more than the number of processors available. This resulted in the need for processing more than one topic on one processor. To reduce message passing between processors, authors define a message buffer, on which the messages are propagated, before being passed.

3.7. Datasets

The available datasets in the scope of the thesis are categorized as datasets for collaborative filtering, content based filtering and social networks. Collaborative filtering datasets are Movielens [73], movie-rating and Netflix [57] datasets. Movielens and movie-rating datasets include a two dimensional matrix where rows correspond to users and columns correspond to movies. Additional user-related information available on Movielens dataset are age, gender, city and postbox. The movie name is the only movie related information given along with this dataset. For movie-rating dataset, only

the 2-D user-movie matrix is given. Netflix dataset includes the features of Movielens dataset but it also provides rating dates which is an important source of knowledge. Another source of information valuable for the recommendation systems domain is item tag data. Based on the vision of semantic web, applications that gather tags for online data has increased dramatically. There are many tag datasets available. Some of these datasets are CiteULike [70], Orkut [71] and Bibsonomy [72]. IMDB tags dataset [91] is used to couple movie ID with other datasets like Netflix and Movielens to provide additional information like actors, comments from IMDB to other datasets. The last category of datasets is the social network data. Epinions [89], CiteULike, Orkut and Bibsonomy datasets include this feature under different forms. CiteULike and Bibsonomy datasets include the social network in the form of citations or cited users whereas Orkut and Epinions datasets have the friendship/trust feature that defines a social network.

3.8. Summary

To sum up, the following requirements should be met by an effective recommender system.

1. *Sparsity*: The high dimensional input data is highly sparse. Correlation calculation on this data is very costly and there is room for representing the real input data in lower dimensions.
2. *Scalability*: A recommender system application should handle the sparse and high dimensional input data and provide rating predictions in real time.
3. *Flexibility*: The method should be able to integrate various data sources that are effective in the rating prediction process.
4. *Explanation ability*: A recommender system should provide qualitative and quantitative reasoning for the rating prediction.

Probabilistic latent semantic analysis satisfies these requirements, because;

- It provides a low dimensional representation of data based on solid statistical background.

- As a model based algorithm, model training can be done offline and a prediction can be provided in constant time online.
- This model is applicable to any two-mode co-occurrence data, which is referred to as dyadic data in the literature. Document-word and user-item co-occurrence relations are examples of the dyadic data concept. So, it is possible to incorporate additional data sources in a unified manner with probabilistic latent semantic analysis.
- Scalability of the probabilistic latent semantic analysis is a studied topic in the literature. The model can adjustably favor accuracy or time complexity.

CHAPTER 4

RECOMMENDATION BY USING NETWORK REGULARIZATION ON GPLSA

4.1. Methodology

We propose a network regularization method on GPLSA [68] and improve this method by the use of trust based methods as the core of the regularization method. The GPLSA model and the network regularization method are explained in this chapter.

The network regularization approach builds on the semi-supervised learning approaches [5], [6], [52] that are explained in Chapter 2. Semi-supervised learning aims to improve clustering algorithms with the incorporation of a small labeled dataset. In recommender systems, a small set of labeled examples are available as observed ratings and recommender systems try to cluster these unobserved ratings. The dyadic nature of datasets on recommender systems requires models to cluster the data on each dimension of the dyad. So, the K-means algorithm is not suitable for the recommendation domain. But the modifications on the expectation maximization algorithm of K-means provide the basis for integrating the semi-supervision into probabilistic latent semantic analysis.

Starting from this analogy, we have investigated how the pairwise constraints between elements of a dimension can be exploited throughout the clustering process, as in the constrained k-means approach.

On the recommendation systems domain, this idea translates into the usage of pairwise local user similarity data throughout the recommendation process. Currently, the usage of social trust network data occurs either as an initialization step or post processing step on the recommendation process. We propose the usage of trust propagation operators as a heuristic for improving the model based collaborative filtering performance.

4.2. Gaussian Extension to PLSA Model – Gaussian PLSA

In this section we give the modified parts of the PLSA model for CF and the reasoning behind the modification. The Gaussian PLSA model is illustrated in Figure 14. The user-item-rating triples are represented by variables <U,Y,V> and hidden variables are represented by variable Z.

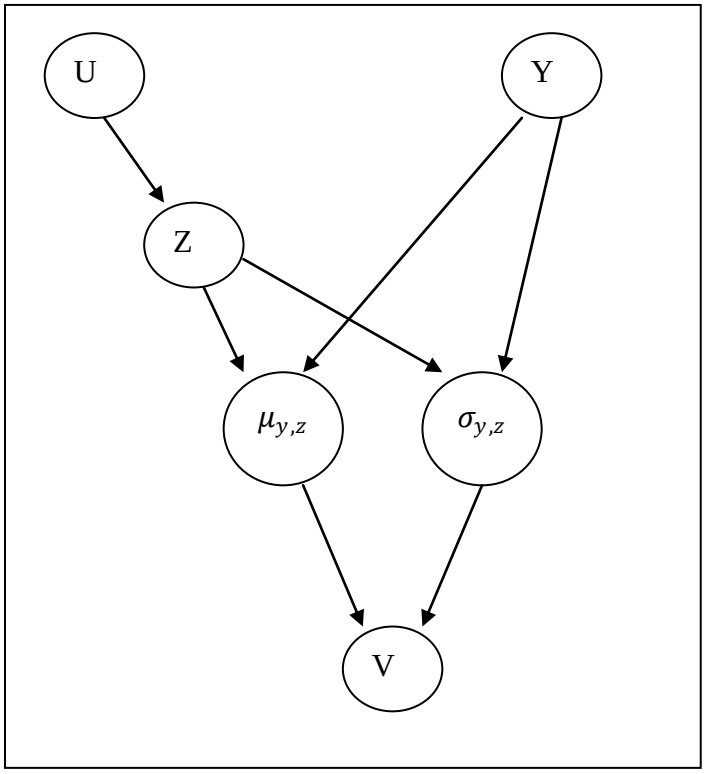


Figure 14 – Gaussian PLSA Model

The first difference of the Gaussian PLSA [68] is that this model considers a risk function and aims to maximize the negative complete risk function instead of the complete data log-likelihood. The likelihood of the Gaussian PLSA is given in equation (4.1).

$$L((u,v,y),\theta) = - \log p(v|u,y;\theta) \quad (4.1)$$

The proposed mixture model is given in equation (4.2).

$$P(v|u,y) = \sum_z p(v|y,z)P(z|u) \quad (4.2)$$

The standard PLSA approach models the co-occurrence of words and documents in a corpus. It has two main drawbacks. The first drawback is that the model does not have the ability to capture the user specific voting scales. The second drawback is that it is not able to respond with an explicit rating value. Different from the standard co-occurrence model, the aim of Gaussian PLSA is to model explicit rating values. The proposed solution in Gaussian PLSA is using topic-user distributions, $P(z|u)$, as user-specific mixture weights of Gaussian parameters where the $p(v|y,z)$ are modeled as Gaussian distributions, $p(v;\mu_{y,z},\sigma_{y,z})$.

$$p(v;\mu,\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{[-\frac{(v-\mu)^2}{(2\sigma^2)}]} \tag{4.3}$$

Given the free parameters of the model $\mu_{y,z},\sigma_{y,z}$ and $P(z|u)$, the expected value of each rating value can be computed by the following formula.

$$\mathrm{E[v|u,y]} = \int_v vp(v|u,y)dv = \sum_z P(z|u)\,\mu_{y,z} \tag{4.4}$$

The log-likelihood function can be written as below.

$$R^c(\theta) = -\sum_{\langle u,v,y,z\rangle} \log p(v|y,z) + \log P(z|u) \tag{4.5}$$

The brackets in the summation represents that the summation is done over the observed values of the variables inside.

In the expectation maximization method of PLSA, the expectation step is dedicated to the calculation of the posterior of the model, $P(z|u,v,y;\theta)$.

$$P(z|u,v,y;\theta) = \frac{p(v|y,z)P(z|u)}{\sum_{z'} p(v|y,z')P(z'|u)} \tag{4.6}$$

The maximization step updates the values of the free parameters on the model. The free parameters are $\mu_{y,z},\sigma_{y,z}$ and $P(z|u)$. The updating schemes for these parameters are given below.

$$P(z|u) = \frac{\sum_{\langle u',v,y\rangle:u'=u} P(z|u,v,y;\theta)}{\sum_{z'}\sum_{\langle u',v,y\rangle:u'=u} P(z'|u,v,y;\theta)} \tag{4.7}$$

$$\mu_{y,z} = \frac{\sum_{\langle u',v,y\rangle:y'=y} vP(z|u,v,y;\theta)}{\sum_{\langle u',v,y\rangle:y'=y} P(z|u,v,y;\theta)} \tag{4.8}$$

$$\sigma_{y,z} = \frac{\sum_{\langle u',v,y\rangle:y'=y}(v-\mu_{y,z})^2 P(z|u,v,y;\theta)}{\sum_{\langle u',v,y\rangle:y'=y} P(z|u,v,y;\theta)} \tag{4.9}$$

The expectation and maximization steps are iterated successively until the Gaussian parameters converge.

The Gaussian PLSA models the probability of each user's membership to each community with $P(z|u)$ value. Item ratings under each community are modeled by Gaussian parameters. Intuitively, the model considers community-wide item ratings and normalizes the rating values according to community mean and variance for that item. But users may have varying voting scales. A user may give hardly a rating of 3 while another may easily give a rating value of 1. For this reason, user normalization is applied before model training and recommendation generation steps. Another reason for user normalization is the different number of votes given by users. User normalization offers to pull seldom raters' rating values to global mean and variance values by a weight.

User normalization is applied on the real rating matrix. After the user normalization step, the rating values are re-scaled between 0 and 1, so that different voting behaviors are represented in a common interval.

We need rating Gaussian parameters, mean and variance, specific to each user. The equations to calculate Gaussian parameters are given below. On these equations, $\bar{\mu}$ and $\bar{\sigma}$ are global rating mean and variance values. The q weight handles the second issue which is the seldom raters. The q value is the weight that controls to which extent the observed rating values should be pulled to the global mean and variance.

$$\mu_u = \frac{\sum_{\langle u',v,y\rangle u'=u} v + q\bar{\mu}}{N_u + q} \tag{4.10}$$

$$\sigma_u^2 = \frac{\sum_{\langle u',v,y\rangle u'=u} (v - \mu_u)^2 + q\bar{\sigma}^2}{N_u + q} \tag{4.11}$$

In equations 4.10 and 4.11, μ_u, σ_u are calculated from rating values. After model training, we acquire item-community Gaussian parameters, $\mu_{y,z}$, $\sigma_{y,z}$ and user community membership probabilities, $P(z|u)$. Finally, the user normalized rating prediction equation 4.12 is used to output the estimated rating values.

$$p(v|u,y) = \mu_u + \sigma_u \sum_z P(z|u)p(v'; \mu_{y,z}, \sigma_{y,z}) \tag{4.12}$$

In this equation, v' rating values are obtained with a simple transformation given in 4.13.

$$v' = \frac{v - \mu_u}{\sigma_u} \tag{4.13}$$

This equation handles the issue related to varying voting scales.

4.3. Network Regularization

The work in [79] proposes the regularization of document hidden semantic space using the document similarity prior. This idea is proposed to overcome the overfitting problem of PLSA. When PLSA overfits to a data which is not sufficiently accurate, regularization techniques help on smoothing the hidden semantic space. In this section, we explain how we incorporate Gaussian PLSA and the method in [79], the Laplacian PLSI which is referred to as LapPLSI.

The aim is to increase the user clustering performance over the probability values $P(z|u)$. Since the posterior of the model is not affected, the expectation step of Gaussian PLSA algorithm stays the same. The modification of the maximization step only considers the $P(z|u)$ value. Also the loglikelihood of the model changes. The

modification of the loglikelihood depends on the chosen regularization function. The chosen regularization function in [79] is the Euclidean distance metric.

The similarity matrix W, given below, is used for smoothing the conditional distribution function $P(z|u)$.

$W_{i,j} = \begin{bmatrix} \cos(u_i, u_j), & if\ u_i \in N_p(u_i)\ and\ u_j \in N_p(u_j) \\ 0, & \text{otherwise} \end{bmatrix}$, where $N_p(u_x)$ represents p nearest neighbor documents of user u_x.

The regularizer function indexed on the hidden topic weighs inter-document topic distances, $\left(P(z_k|u_i) - P(z_k|u_j)\right)^2$, with their cosine distance, W_{ij}.

$$
\begin{aligned}
R_k &= \frac{1}{2}\sum_{i,j=1}^{N}\left(P(z_k|u_i) - P(z_k|u_j)\right)^2 W_{ij} \\
R_k &= \sum_{i=1}^{N}\left(P(z_k|u_i)\right)^2 D_{ii} + \sum_{i,j=1}^{N} P(z_k|u_i)P(z_k|u_j)W_{ij} \\
R_k & \\
&= [\,P(z_k|u_1) \dots P(z_k|u_n)\,]\ D_{ii}\ [\,P(z_k|u_1) \dots P(z_k|u_n)\,]^T \\
&- [\,P(z_k|u_1) \dots P(z_k|u_n)\,]\ W_{ij}\ [\,P(z_k|u_1) \dots P(z_k|u_n)\,]^T \\
R_k &= [\,P(z_k|u_1) \dots P(z_k|u_n)\,]\,(D - W)\ [\,P(z_k|u_1) \dots P(z_k|u_n)\,]^T \\
R_k &= [\,P(z_k|u_1) \dots P(z_k|u_n)\,]\,L\ [\,P(z_k|u_1) \dots P(z_k|u_n)\,]^T
\end{aligned}
\tag{4.14}
$$

$L = D - W$ is called graph laplacian. The regularization applied on the log-likelihood of the model is as shown in equation 4.15.

$$
\begin{aligned}
L_R &= L - \lambda\sum_k R_k \\
L_R &= \sum_{i=1}^{N}\sum_{j=1}^{M} n(u_i, i_j)\log\sum_{k=1}^{K} P(i_j|z_k)P(z_k|u_i) \\
&\qquad -\frac{\lambda}{2}\sum_{k=1}^{K}\sum_{i,j=1}^{N}\left(P(z_k|u_i)P(z_k|u_j)\right)^2 W_{ij}
\end{aligned}
\tag{4.15}
$$

λ parameter controls the effect of the regularization. The generalized expectation maximization algorithm for probabilistic latent semantic analysis repeats the maximization step until the log-likelihood decreases. When the log-likelihood decreases, the parameter values are set to their previous values which defines a local optimum and the generalized EM algorithm continues with the expectation step.

Unlike the regular EM algorithm of PLSI, LapPLSI uses a variation of EM algorithm, generalized expectation maximization. The generalized EM algorithm for LapPLSI is as follow:

- Expectation Step
  - Calculate P(z|d,w) using P(z), P(z|d) and P(w|z).
- Maximization Step
  - Calculate P(z) and P(w|z) from P(z|d,w).
  - Regularization Step (Generalized EM modification)
    - Calculate regularized P(z|d) .
    - Calculate inner log-likelihood.
    - If inner log-likelihood > general log-likelihood
      - general log-likelihood = inner log-likelihood.
      - P(z|d) = regularized P(z|d).
      - go to halt condition step.
    - Else go to regularization step.
- Check halt condition
  - Calculate general log-likelihood of the model.
  - If general log-likelihood change is less than a threshold,
    - halt EM steps.
  - Else go to expectation step

The calculation of the regularized $P(z|u)$, $P_{reg}(z|u)$, is given in equation 4.16.

$$P_{reg}(z_k|u_i) = (1-\gamma)P(z_k|u_i) + \gamma \frac{\sum_{j=1}^{N} W_{ij} P(z_k|u_j)}{\sum_{j=1}^{N} P(z_k|u_j)} \tag{4.16}$$

4.4. Network Regularization on Gaussian PLSA

The user community based item distances represent both the rating patterns of the user and the local social trust network of the user in the network regularization on Gaussion

PLSA. Two users, who have significantly different rating patterns but high trust in each other, converge to each other in the semantic space, formed by $P_{reg}(z|u_i)$.

The regularization scheme of Gaussian PLSA is different from the original PLSA model. The user topics and item Gaussian parameters are dependent on each other. So, the regularization on user topics using the trust network requires updating of item Gaussian parameters. We have implemented the following regularization scheme for the Gaussian PLSA model.

1. Select a relevant set of users for the active user and regularize active user's topics with a weighting scheme.
2. Update Item Gaussian parameters using the mean (eq. 4.17) and variance (eq. 4.18) of the ratings of only the specified set of users.

$$\mu_{yz}^{t+1} = \beta * \mu_{yz}^{t} + (1-\beta) * \frac{\sum_{u \in trustSet} v_{u,y}}{|trustSet|} \tag{4.17}$$

$$\sigma_{yz}^{t+1} = \beta * \sigma_{yz}^{t} + (1-\beta) * \frac{\sum_{\langle u,v,y' \rangle : y'=y, u \in trustSet} \left(v - \mu_{y,z}^{t}\right)^2}{|trustSet|} \tag{4.18}$$

The first step of the regularization scheme can be applied with any weighting method. This provides the freedom to choose any trust based method.

The modified training algorithm of Gaussian PLSA for network regularization is as follows.

1. Iterate the Gaussian PLSA EM algorithm once and obtain parameter set, P_1.
2. Update parameters with the regularization scheme and obtain P_2.
3. Iterate the Gaussian PLSA EM algorithm with parameters P_2 once and obtain parameter set P_3.
4. If the likelihood increases with parameters P_3,
   set P_1=P_3, increase counter.
5. If the likelihood doesn't increase,
   go to 3.
6. If counter < Max. no. of iterations, halt.
   Else go to 1.

4.5. Usage of Local Trust Metrics as Weighting Schemes

The state of the art trust enhanced collaborative filtering methods are compared in [81]. The trust based recommendation algorithms differ in the trust propagation method and whether they incorporate the similarity of rating patterns.

The trust based methods in [81] assume that the trusted users' ratings are close to the trusting user's ratings. So, these methods define a trusted user set for each user. The set is acquired by walking over the closest trusted users to transitively trusted users. The depth of the walk is limited for performance concerns. The trust weight of the active user to a trusted user is maintained by the length of the path between the two users and the weight is diminished by each step on the graph.

For the active user, TidalTrust [87] predicts an unobserved item's rating by finding the closest users in the trust graph and averaging the user ratings in the path weighted by each user's distance to the active user.

Moletrust [88] differs from TidalTrust in its calculation of weight values and the stopping condition of the walk. The graph search is applied until a depth that is initially specified. The calculation of the trust weight values not only considers the path length but also the trusting users of the node in each step.

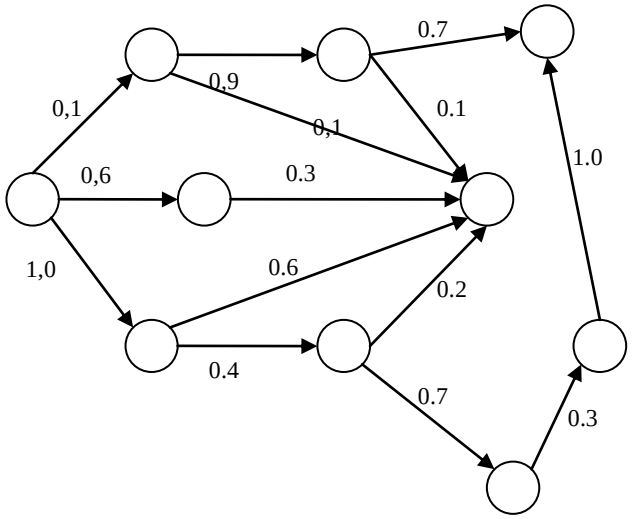


Figure 15 – Trust network example

Figure 15 illustrates a trust network. MoleTrust has a predefined maximum depth value for breadth first search. In the example, the maximum depth value is 2. As a local trust metric, the algorithm takes one of the users as the active user and applies the breadth first search.

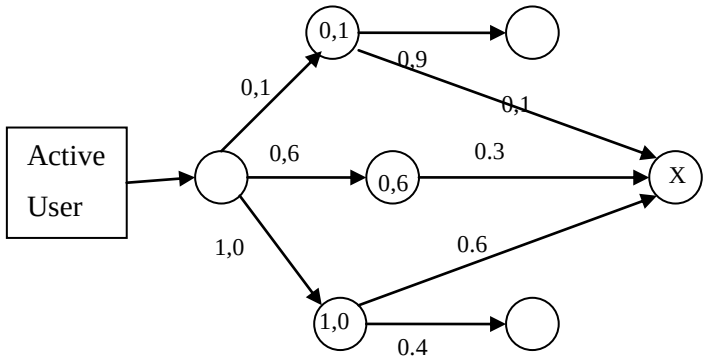


Figure 16 – MoleTrust example, walk depth=2, threshold>0.1

Figure 16 shows the subgraph for the selected active user with maximum depth 2. The trust weights for the users that are directly connected to the active user are the corresponding edge weights. MoleTrust assesses the trust weight between the active user and user X using equation 4.19.

$$MoleTrust_{active\ -X} = \frac{\sum_{paths\ P} \prod_{edges\ \ e\ in\ path\ p \in P} weight_e}{\sum_{paths\ p} first\ edge\ weight\ in\ p} \tag{4.19}$$

For our sample network, the trust value between the active user and user X is calculated as in equation 4.20. Notice that the path starting with edge weight less than or equal to the threshold is pruned. Final moletrust subgraph with similarity assessment between active user and destionation user is illustrated in Figure 17.

$$MoleTrust_{active\ -X} = \frac{0{,}6 * 0{,}3 + 1{,}0 * 0{,}6}{0{,}6 + 1{,}0} = 0{,}4875 \tag{4.20}$$

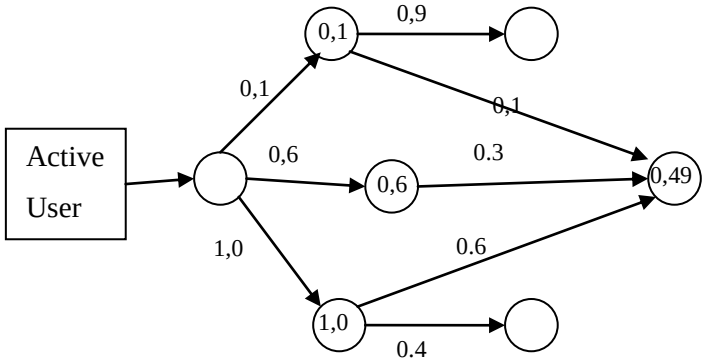


Figure 17 – MoleTrust example, trust weight between active user and X

Supposing user X is the destination, the final subgraph for TidalTrust algorithm is given in Figure 18.

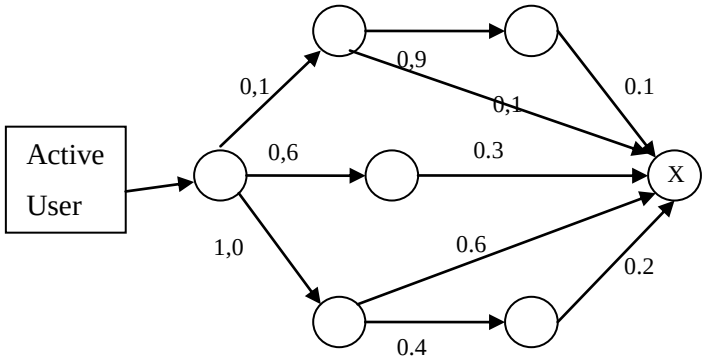


Figure 18 – TidalTrust example

The biggest difference of TidalTrust algorithm from MoleTrust is the propagation method. For a destination node, TidalTrust algorithm uses shortest paths and does not have a predefined maximum depth value. For our example, the calculation of the trust weight between the active user and X is given in equation 4.21. Notice that paths of length 3 are not considered on the weight calculation.

$$TidalTrust_{active\ -X} = \frac{0{,}6 * 0{,}3 + 1{,}0 * 0{,}6 + 0{,}1 * 0{,}1}{0{,}6 + 1{,}0 + 0{,}1} = 0{,}4906 \qquad (4.21)$$

[81] also proposes the state of the art trust based methods for trust enhanced collaborative filtering. The corresponding trust enhanced recommendation algorithms are given in equations 4.22 and 4.23. R^T denotes users in the trust set. The trust set is obtained by the corresponding algorithm in the following equations. $t_{a,u}$ is the trust weight between users *a* and *u*. $w_{a,u}$ denotes the rating pattern similarity of users *a* and *u*.

TidalTrust based recommendation algorithm is referred as the trust based weighted mean in [81]. The trust set and trust weights are obtained by the TidalTrust propagation method.

$$TidalTrust - v_{a,i} = \frac{\sum_{u \in R^T}^{k} t_{a,u} v_{u,i}}{\sum_{u=1}^{k} t_{a,u}} \tag{4.22}$$

The recommendation algorithm based on MoleTrust exchanges the similarity weight in PCC with trust weights.

$$MoleTrust - v_{a,i} = \overline{v_a} + \frac{\sum_{u \in R^T}^{k} t_{a,u} (v_{u,i} - \overline{v_u})}{\sum_{u=1}^{k} t_{a,u}} \tag{4.23}$$

Another trust based recommendation is the Trust Based Filtering (TBF) method. The difference of this algorithm from the MoleTrust enhanced CF algorithm is that this algorithm expands the trusted users set with the users who have a similarity value above a predefined threshold. TBF method does not provide a propagation operator and it uses direct connections in the network to assess the trusted users set.

$$TBF - v_{a,i} = \overline{v_a} + \frac{\sum_{u \in R^{T+}}^{k} t_{a,u} (r_{u,i} - \overline{v_u})}{\sum_{u=1}^{k} t_{a,u}} \tag{4.24}$$

In summary, our main work is divided in three parts which successively build onto each other. The first part of our work is the Gaussian extension of the PLSA approach as the model based solution for the collaborative filtering problem. The second part is the regularization approach on GPLSA. In this part, the regularization approach for GPLSA is proposed which is original to this thesis. The third part is the usage of available trust based heuristics as the propagation operator for regularization.

CHAPTER 5

EXPERIMENTS AND EVALUATION

This chapter is dedicated to the presentation of the methodology and results of the experiments. The methodology includes the experimentation settings, dataset specifications and the evaluation method. The experimentation is divided into three subsections. In the first subsection, the model based and memory based algorithm performances are compared. In the second subsection, performance results for the trust based methods are given. The third subsection is the overall comparison of the sole collaborative filtering methods, sole trust based methods and usage of trust based heuristics for network regularization on GPLSA.

5.1. Experimentation Settings

The recommendation process for the model based algorithms is shown in Figure 19. The model based algorithms are Gaussian PLSA and Gaussian PLSA with network regularization methods.

Figure 19 – Model based recommendation process

The recommendation process in Figure 20 is applied for the memory based algorithm, item based TopN recommendation algorithm.

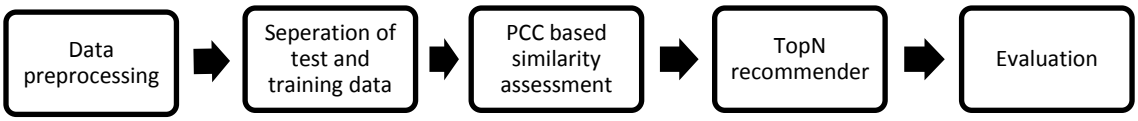


Figure 20 – Memory based recommendation process

We applied p-core processing at the data preprocessing step. The inputs to this step are user-item and user-user matrices. This operation aims to partition the dataset so that at least each user and each item have at least p occurences. Then only the selected users' trust values are extracted from the user-user matrix.

The next step is the addition of noise to data. In this step, we randomly separate values from the dataset and set the separated variables as unobserved in the original matrix. To achieve this, for each user u_i, we randomly pick K items $i_{rand\,1 \ldots randK}$ which have a declared vote by the user and remove that vote information, $v_{i,\text{rand } 1 \ldots \text{randK}}$.

An optional step before model training is the user normalization. In this optional step, users are normalized around their mean. The user normalization is applied for the different voting behaviours of users. The user normalization is explained in more detail in Section 4.2.

Once the data is preprocessed and the test and training sets are obtained, the process continues with the model training step.

The recommendation step requires the calculation of the rating probabilities given the user and the item. The output of this step are the estimated rating values for the seperated user-item matrix cells.

The final step is the evaluation of recommendation results. The results are evaluated using minimum absolute error.

5.2. Performance Evaluation

The performance evaluation step takes the real and predicted ratings, and outputs an overall recommendation accuracy. Our evaluation is the prediction accuracy on numerical rating values which are in a common range.

The method used for measuring the accuracy of explicit rating predictions is the minimum absolute error (MAE) which measures the mean of absolute difference between the actual (y_i) and predicted ($f(x_i)$) rating values.

$$MAE = \sqrt{\frac{\sum|f(x_i) - y_i|}{n}} \tag{5.1}$$

5.3. Sole CF Experiment

In this section, we experiment the model based collaborative filtering method, Gaussian PLSA, for the prediction of explicit ratings. The collaborative filtering algorithms are the memory based algorithm PCC, and the model based algorithm GPLSA.

5.3.1. Sole CF Experiment Dataset

Epinions [89] is a social networking site where users can comment on any product. It provides trust values between each user. There are many types of products like movies, songs, hardwares, softwares, appliances that can be commented on, which increase the sparsity. The dataset includes user-item rating data and user-user trust data. The rating data is composed of $user_i$-$item_j$-$rating_n$ triples defining the rating value $rating_n$ that $user_i$ has given to $item_j$. The preprocessed dataset has 46914 users and 100922 items. The rating matrix has 523k rating statements. Rating value is an integer between 1 and 5. The sparsity of rating matrix is:

$$Sparsity_{Epinions} \cong \frac{523k}{47k * 101k} = 4{,}34 * 10^{-4} \tag{5.2}$$

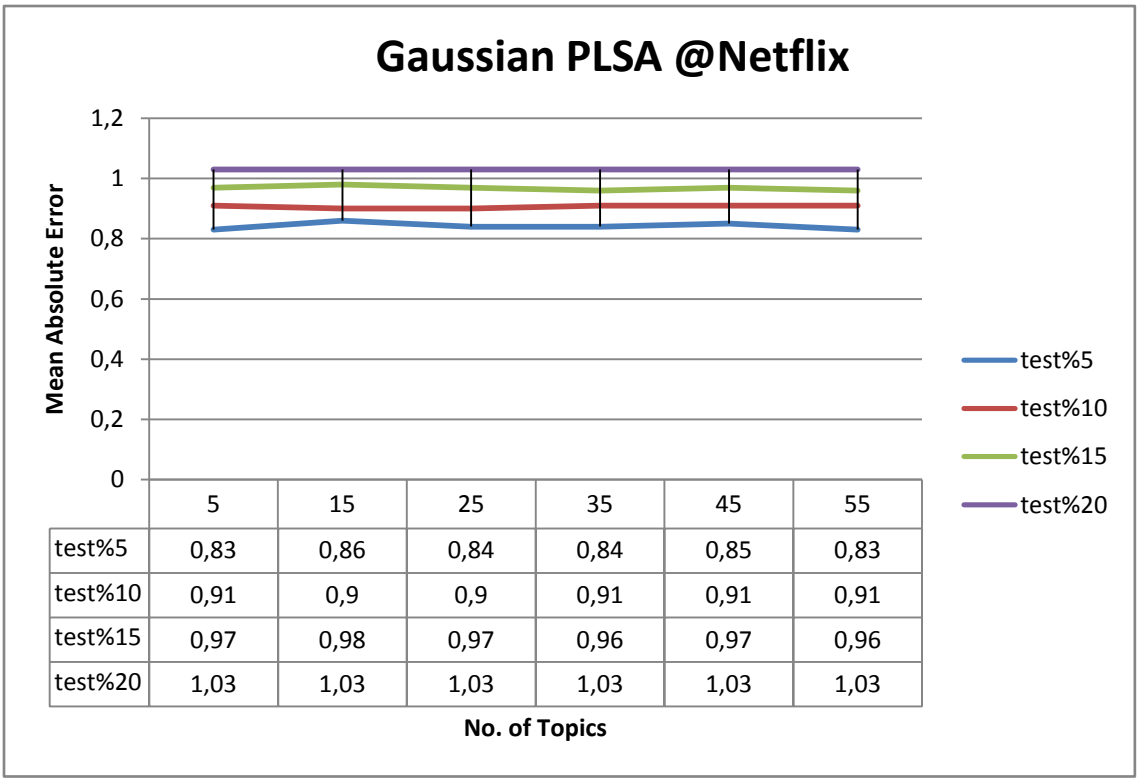
The sole CF experiments require only the rating matrix of the Epinions dataset. The rating matrix of the Epinions dataset is called *Epinions-Rating* in the experiment results.

The Netflix [90] dataset is also used to compare a multi-domain user-item matrix and a single domain user-item matrix.

5.3.2. Sole CF Experiment Results

Sole CF experiment includes

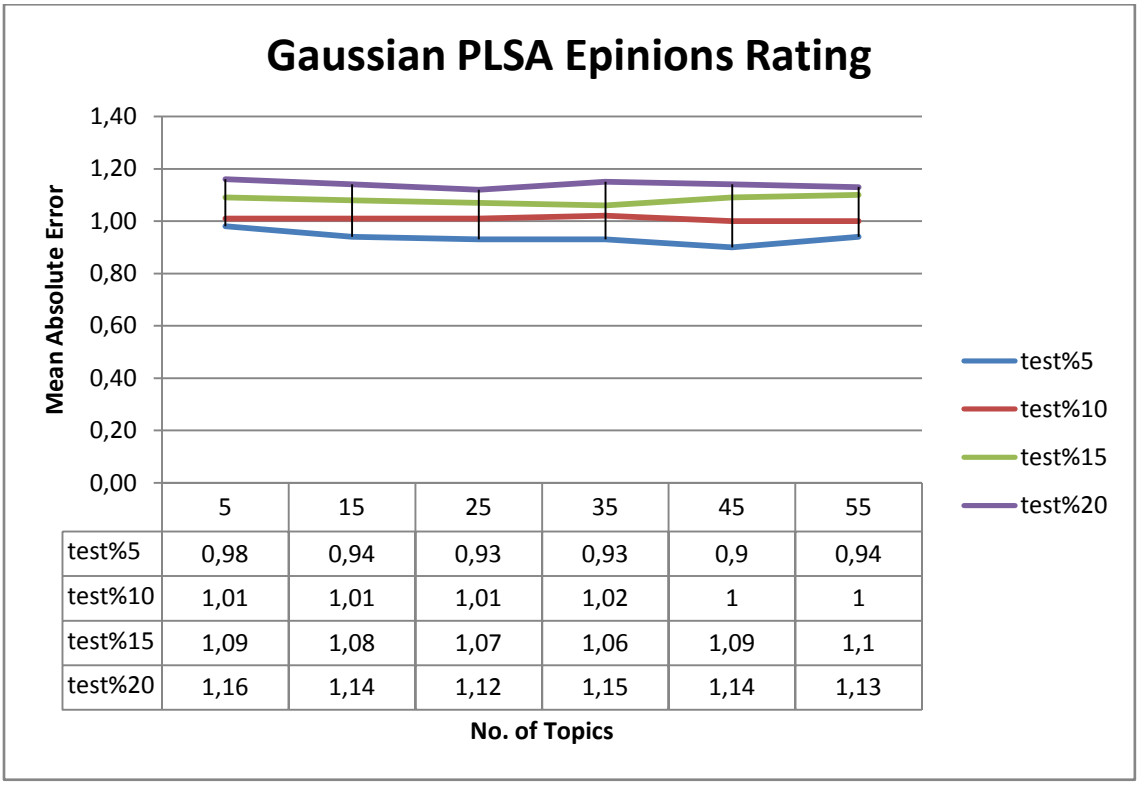
- Results for GPLSA on Netflix and Epinions
- Results for PCC on Netflix and Epinions
- Comparison on GPLSA and PCC on Netflix and Epinions



	5	15	25	35	45	55
test%5	0,83	0,86	0,84	0,84	0,85	0,83
test%10	0,91	0,9	0,9	0,91	0,91	0,91
test%15	0,97	0,98	0,97	0,96	0,97	0,96
test%20	1,03	1,03	1,03	1,03	1,03	1,03

Figure 21 – The effect of noise level and number of topics on GPLSA@Netflix

Figure 21 shows the performance of Gaussian PLSA performance on the Netflix dataset. The number of topics does not affect the performance of the algorithm significantly. As expected, MAE increases with the addition of noise, since the noise increases entropy.

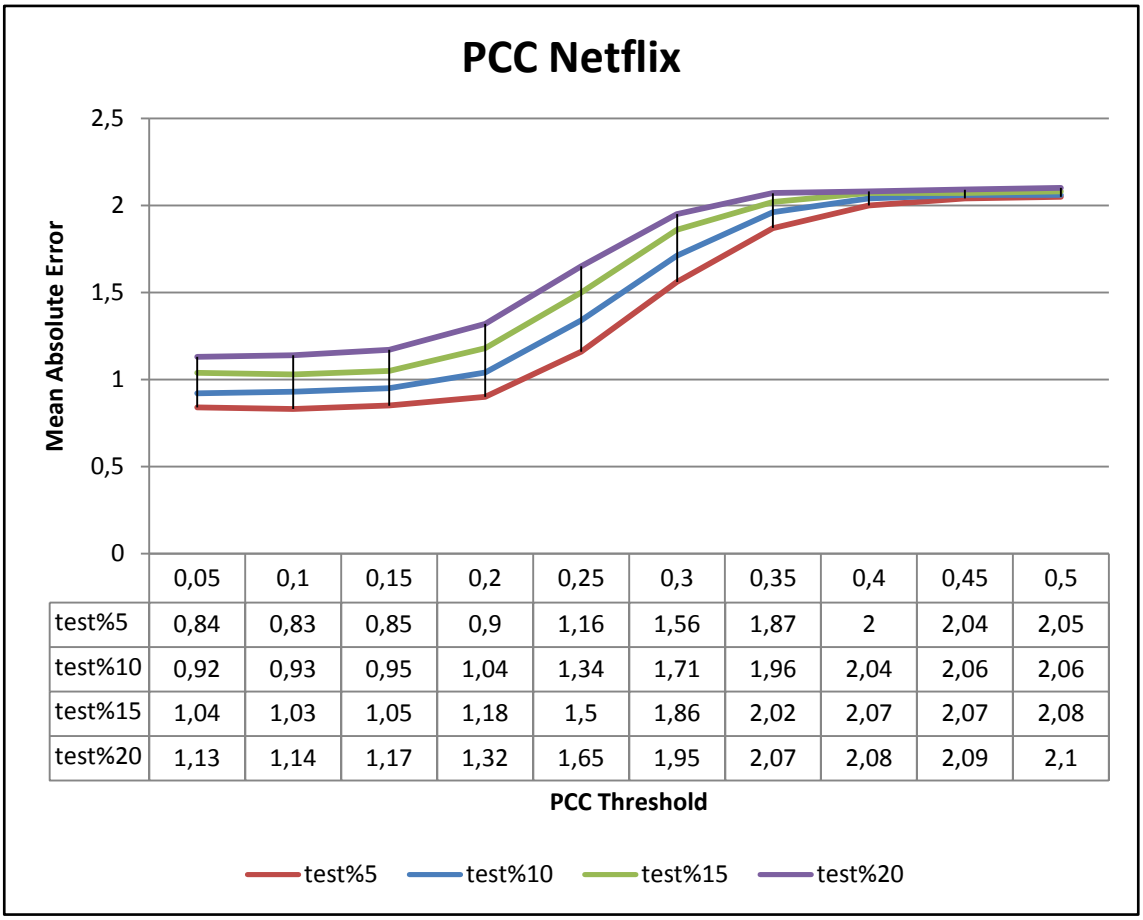


	5	15	25	35	45	55
test%5	0,98	0,94	0,93	0,93	0,9	0,94
test%10	1,01	1,01	1,01	1,02	1	1
test%15	1,09	1,08	1,07	1,06	1,09	1,1
test%20	1,16	1,14	1,12	1,15	1,14	1,13

Figure 22 – Effect of noise level and number of topics on GPLSA@Epinions-Rating

In Figure 22, the Gaussian PLSA is applied on the user-item matrix of the Epinions dataset. MAE results of the Gaussian PLSA on Epinions dataset are higher than the experiment with the Netflix dataset. The main reason behind this result is the sparseness of Epinions dataset. Another reason for the diminished performance is that users may have varying rating policies for items under two different categories. As stated in 5.3.1, the Epinions dataset is a multi-domain dataset which provides various items from various categories. So, a user can give ratings for items in electronics category with mean 3 and variance 1.1, while the ratings observed for the items in books category are with mean 4 and variance 0.5. This case can also be observed for the different film categories but it is more likely in a dataset of multi domain.

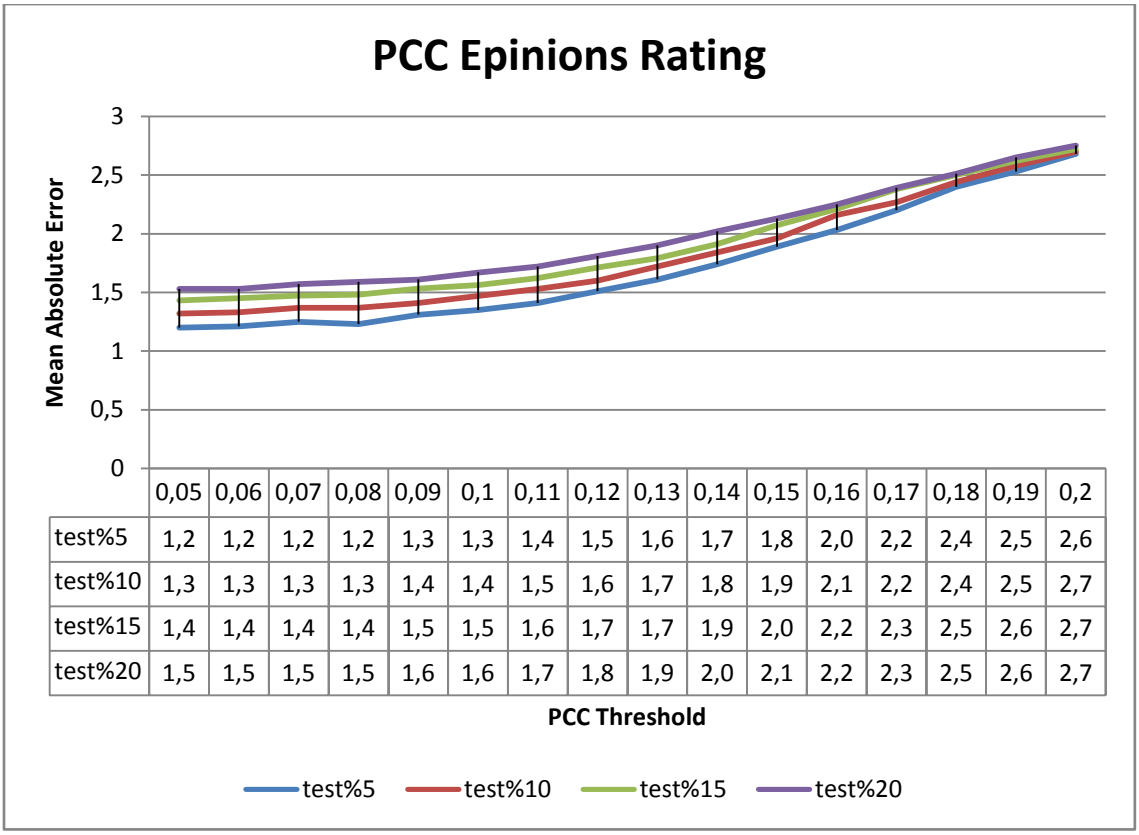
We also implemented PCC algorithm as a memory based algorithm and a baseline for our experiments. We observed the effect of noise and similarity thresholds of PCC on both Netflix and Epinions-rating datasets with MAE as the performance metric.



	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
test%5	0,84	0,83	0,85	0,9	1,16	1,56	1,87	2	2,04	2,05
test%10	0,92	0,93	0,95	1,04	1,34	1,71	1,96	2,04	2,06	2,06
test%15	1,04	1,03	1,05	1,18	1,5	1,86	2,02	2,07	2,07	2,08
test%20	1,13	1,14	1,17	1,32	1,65	1,95	2,07	2,08	2,09	2,1

Figure 23 - Effect of noise level and similarity threshold on PCC@Netflix

We observed that when PCC threshold is lower than 0.05, the algorithm cannot find neighbors for most of the users. So, we started the experiments with 0.05 similarity threshold. It is also observed in Figure 23 that the performance decreases after 0.2 similarity threshold for all noise levels.

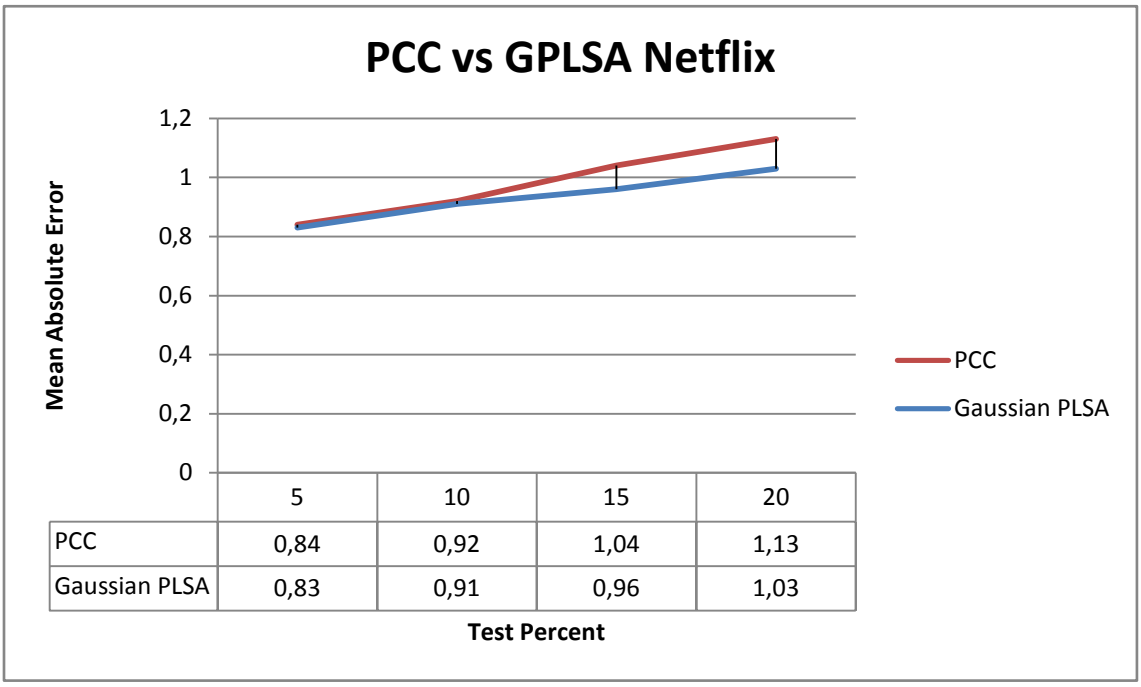


	0,05	0,06	0,07	0,08	0,09	0,1	0,11	0,12	0,13	0,14	0,15	0,16	0,17	0,18	0,19	0,2
test%5	1,2	1,2	1,2	1,2	1,3	1,3	1,4	1,5	1,6	1,7	1,8	2,0	2,2	2,4	2,5	2,6
test%10	1,3	1,3	1,3	1,3	1,4	1,4	1,5	1,6	1,7	1,8	1,9	2,1	2,2	2,4	2,5	2,7
test%15	1,4	1,4	1,4	1,4	1,5	1,5	1,6	1,7	1,7	1,9	2,0	2,2	2,3	2,5	2,6	2,7
test%20	1,5	1,5	1,5	1,5	1,6	1,6	1,7	1,8	1,9	2,0	2,1	2,2	2,3	2,5	2,6	2,7

Figure 24 - Effect of noise level and similarity threshold on PCC@Epinions-Rating

In Figure 24, we tested PCC results on Epinions-Rating matrix. We have stopped the experiment on 0.2 similarity threshold since the greater similarity values do not provide the relevant information for our experiment.

In Figure 25 and Figure 26, we compare the results of the two CF algorithms.



	5	10	15	20
PCC	0,84	0,92	1,04	1,13
Gaussian PLSA	0,83	0,91	0,96	1,03

Figure 25 – Comparison of PCC and GPLSA at Netflix dataset

In Figure 25, it is observed that GPLSA outperforms PCC algorithm on all noise levels. The performance significance increases with the addition of noise, while the performance difference is insignificant in 5% and 10% noise levels. The outcome shows the sparsity handling capability of our model based CF algorithm GPLSA.

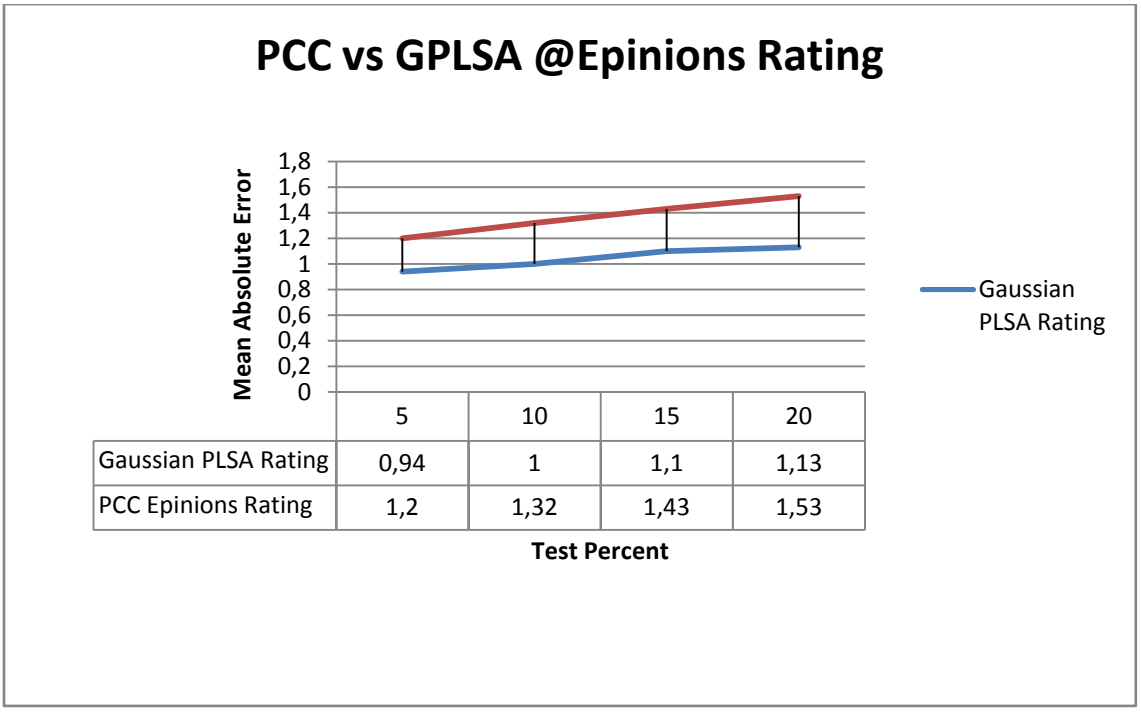
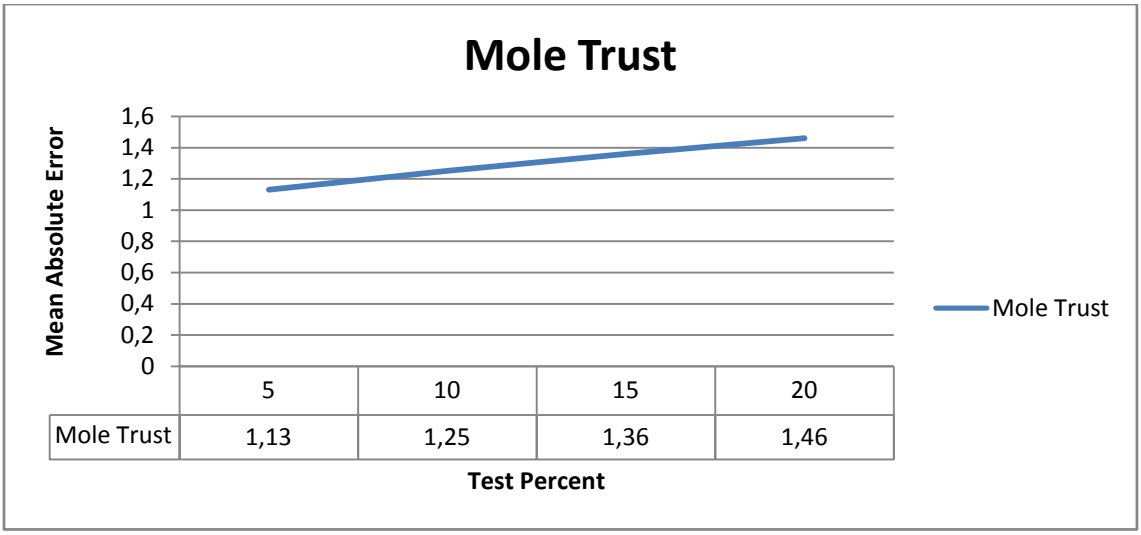


Figure 26 - Comparison of PCC and GPLSA at Epinions-Rating

On Epinions-Rating dataset,which is sparser than Netflix, the sparsity handling capability difference between the two algorithms is observed more clearly (Figure 26).

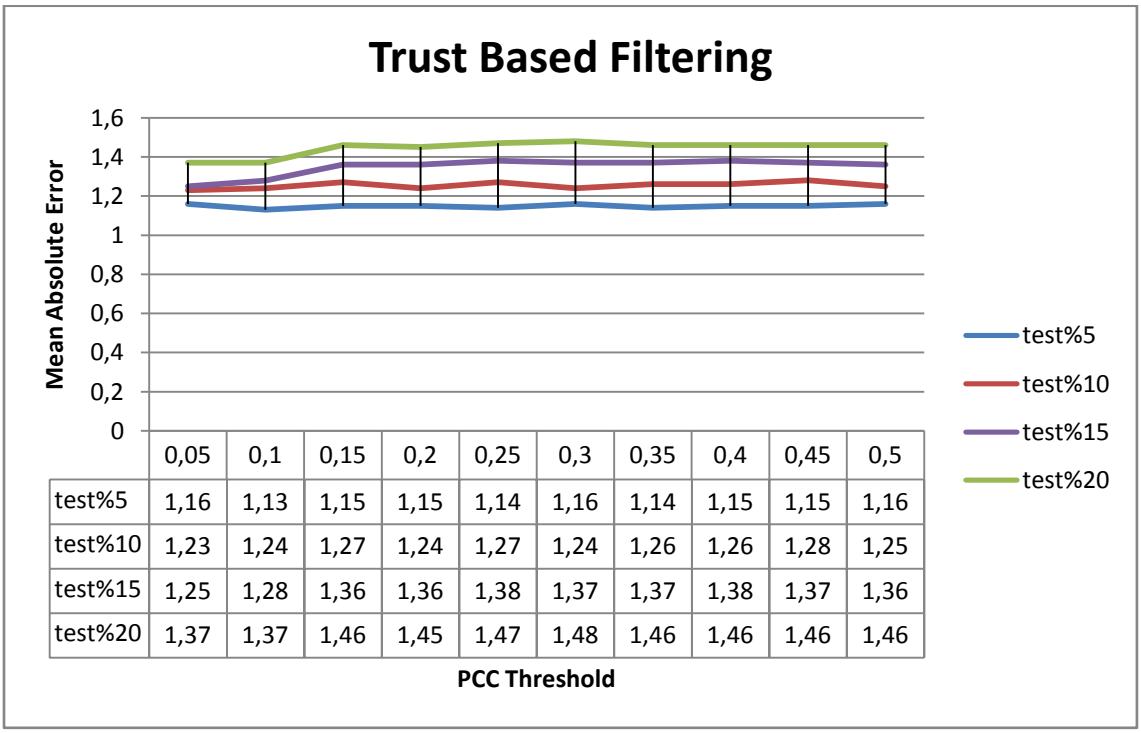
5.4. Trust Enhanced CF Methods

In this part of the experiment, we show the results of the implemented trust based methods. MoleTrust algorithm performs a breadth first search for each user. We observed that MoleTrust performs best at depth 3, so we set the depth to 3 in our experiments. The trust based filtering (TBF) algorithm uses similarity values assessed by PCC. So, the trust based filtering experiment includes the similarity threshold as a free parameter. TidalTrust algorithm is only experimented with respect to the noise ratio. This method, unlike MoleTrust and TBF, does not have an open parameter that should be experimentally set. The experiment results for trust enhanced CF methods are given in Figure 27, Figure 28 and Figure 29.



	5	10	15	20
Mole Trust	1,13	1,25	1,36	1,46

Figure 27 – Effect of noise on MoleTrust algorithm performance



	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
test%5	1,16	1,13	1,15	1,15	1,14	1,16	1,14	1,15	1,15	1,16
test%10	1,23	1,24	1,27	1,24	1,27	1,24	1,26	1,26	1,28	1,25
test%15	1,25	1,28	1,36	1,36	1,38	1,37	1,37	1,38	1,37	1,36
test%20	1,37	1,37	1,46	1,45	1,47	1,48	1,46	1,46	1,46	1,46

Figure 28 – Effect of noise and similarity threshold on Trust Based Filtering algorithm

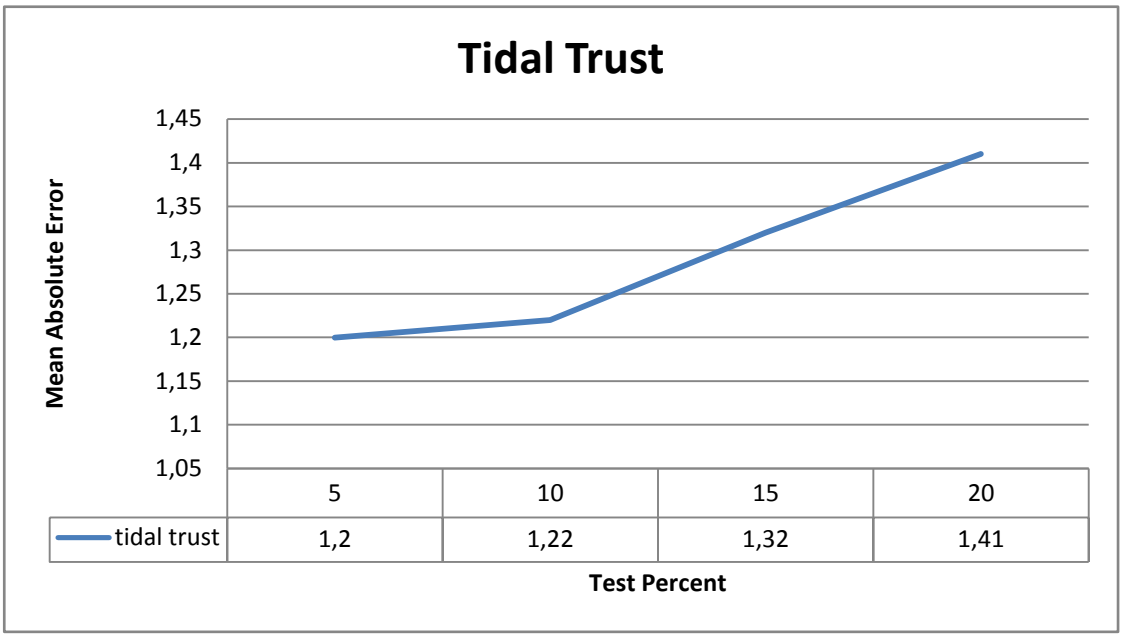


Figure 29 – Effect of noise level on TidalTrust algorithm

5.5. Network Regularization Experiment

The network regularization is applied on the Gaussian PLSA algorithm. The state of the art trust based methods are applied as the core of the network regularization algorithm.

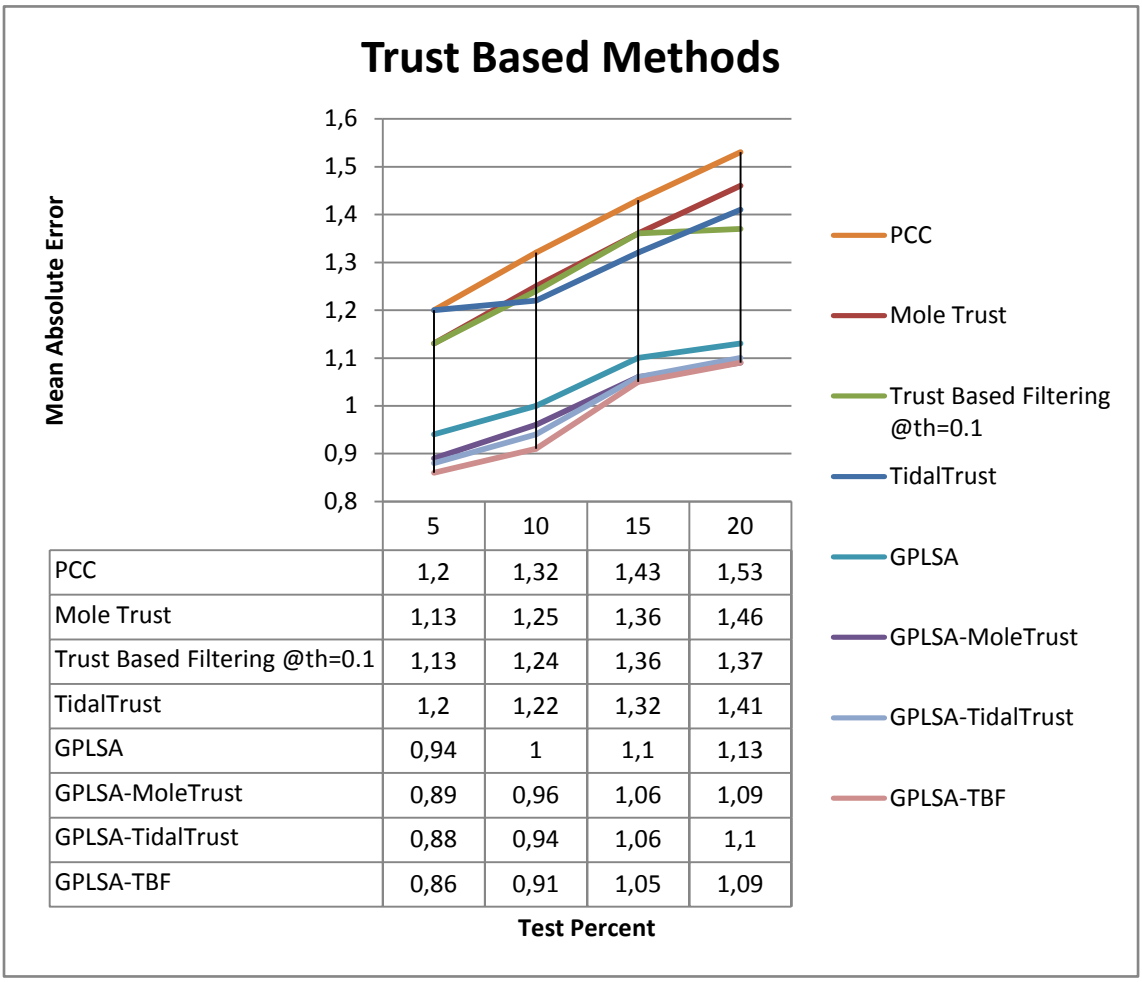
5.5.1. Network Regularization Dataset

For this experiment, both the trust data and the rating data are used from the Epinions dataset. The trust network is used as an adjacency matrix in this experiment. We use the same rating matrix and obtain the user-user trust matrix where users are the same set of users in the rating matrix.

5.5.2. Network Regularization Experiment Results

In this experiment, an overall comparison of the methods given in the previous experiments and the network regularization method is given. For the trust network regularization method, the β parameter which operates as the blending factor between the previous parameter values and the trust enhanced parameter values, is taken as 0,8 (See Section 4.3). The similarity thresholds for the Trust Based Filtering algorithm and

the Trust Based Filtering enhanced network regularization method are taken as 0,1 which is the best experimental value.



	5	10	15	20
PCC	1,2	1,32	1,43	1,53
Mole Trust	1,13	1,25	1,36	1,46
Trust Based Filtering @th=0.1	1,13	1,24	1,36	1,37
TidalTrust	1,2	1,22	1,32	1,41
GPLSA	0,94	1	1,1	1,13
GPLSA-MoleTrust	0,89	0,96	1,06	1,09
GPLSA-TidalTrust	0,88	0,94	1,06	1,1
GPLSA-TBF	0,86	0,91	1,05	1,09

Figure 30 – Overall comparison of methods

5.6. Summary

We firstly experimented the benefit of using model based GPLSA algorithm compared to a memory based algorithm. Figures 25 and 26 show the results of the first experiment on two datasets with different sparsity levels. It is seen that as the sparsity level increases with dataset and addition of noise, the performance of memory based method, GPLSA, decreases significantly less than the memory based baseline. In the second experiment we provided the results of our local trust metrics used to enhance collaborative filtering. The third experiment provides the overall comparison, shown in Figure 30. In this figure, the memory based and model based collaborative filtering

methods, trust enhanced collaborative filtering methods and trust enhanced GPLSA regularization methods are compared. It is observed that the trust propagators are beneficial for regularizing GPLSA model parameters, which proves the main claim of this thesis.

CHAPTER 6

CONCLUSION

In this work, a method for the usage of collaborative and social trust data is proposed. The method is based on a variant of the PLSA model, namely Gaussian PLSA [67]. Network regularization efforts on PLSA model [5, 6, 25, 28, 29, 30, 32] are examined and a method for network regularization on the Gaussian PLSA model is proposed. With these improvements, it is shown that a model based collaborative filtering algorithm, Gaussian PLSA, can be accurately guided by the trust networks in a lower dimension level.

In the first part of the experiments, the performance of a memory based algorithm based on the PCC metric is compared with the model based algorithm, Gaussian PLSA. The experiment is repeated on both the Epinions-Rating dataset and the Netflix dataset. It is observed that Gaussian PLSA always outperforms the baseline memory based algorithm. Another outcome of this experiment is the difference of the two methods in their sparsity handling capabilities. Gaussian PLSA is affected less from the addition of noise in both datasets compared to the memory based algorithm. On the Epinions-Rating dataset which is sparser than the Netflix dataset, the difference in performance between the two algorithms is significantly higher. As a result, it is seen that the model based approach is less prone to sparsity, which is one of the most important problems in recommender systems.

In the second part of the experiments, the results of the state of the art trust enhanced recommendation algorithms, MoleTrust, TidalTrust and Trust Based Filtering, as pointed out in [82], are tested. The Trust Based Filtering algorithm depends on a rating pattern similarity weight which is an additional parameter compared to MoleTrust and

TidalTrust. These methods are used as the trust based baseline methods in the overall experiments.

Our final contribution, which is the choice of network regularization method, is tested using the state of the art local trust metrics as the core of the Gaussian PLSA network regularization method. The trust based filtering heuristic for GPLSA regularization is observed to improve the performance of GPLSA the most.

There are many directions of research in which this thesis can be improved. The improvement opportunities are based on the choice of the probabilistic model, network regularization method for new probabilistic models and choice of different local trust metrics.

The PLSA based model, Gaussian PLSA, is the basis of this thesis. The CF performance can be further improved by implementing the network regularization methods on other PLSA variants for the CF problem. To achieve this, the corresponding model's network regularization method needs to be implemented.

Another possibility of improvement lies in using alternative base models to PLSA. Latent Dirichlet Allocation (LDA) model [13], which provides control on the risk of overfitting problem, is also a powerful technique and an alternative to PLSA model. But the theory behind LDA is different compared to the PLSA model and it requires a completely different network regularization method for parameter updating.

Another dimension for our future work is the experimentation of other local trust metrics. Recent work on random walk methods for trust based collaborative filtering can be an interesting approach to consider for our network regularization method, although the random walk methods are proposed as a global trust metric.

REFERENCES

[1] Hofmann, T. 2001. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning Journal* vol. 42, issue 1/2 (Jan. 2001), pp. 177-196.

[2] Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. 2001. Constrained K-means Clustering with Background Knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning* (June 28 - July 01, 2001). C. E. Brodley and A. P. Danyluk, Eds. Morgan Kaufmann Publishers, San Francisco, CA, pp. 577-584.

[3] Koren, Y. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Paris, France, June 28 - July 01, 2009). KDD '09. ACM, New York, NY, pp. 447-456. DOI= http://doi.acm.org/10.1145/1557019.1557072.

[4] Cohn, D., & Hofmann, T. (2001). The missing link – a probabilistic model of document content and hyper-text connectivity. In *Advances in Neural Information Processing Systems*, Vol. 13. The MIT Press, pp. 430-436.

[5] Xue, G., Dai, W., Yang, Q., and Yu, Y. 2008. Topic-bridged PLSA for cross-domain text classification. In *Proceedings of the 31st Annual international ACM SIGIR Conference on Research and Development in information Retrieval*, SIGIR '08. ACM, New York, NY, pp. 627-634.

[6] K. Zhou, X. G. Rong, Q. Yang, and Y. Yu, Learning with positive and unlabeled examples using topic sensitive PLSA. *IEEE Transactions on Knowledge and Data Engineering*, Jan. 2010, vol. 22, issue 1, pp. 46-58 OAuth, software API for open authentication, http://oauth.net/, Last visited on 10.01.2010.

[7] OpenAuth, software API for open authentication, http://dev.aol.com/api/openauth, Last visited on 10.01.2010

[8] Nielsen Global Online Consumer Survey, http://blog.nielsen.com/nielsenwire/wp-content/uploads/2009/07/pr_global-study_07709.pdf Last visited on 10.01.2010.

[9] Pennock, D. M., Horvitz, E., Lawrence, S., and Giles, C. L. 2000. Collaborative Filtering by Personality Diagnosis: A Hybrid Memory and Model-Based Approach. In *Proceedings of the 16th Conference on Uncertainty in Artificial intelligence* (June 30 - July 03, 2000). C. Boutilier and M. Goldszmidt, Eds. Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers, San Francisco, CA, pp. 473-480.

[10] S. Basu, A. Banerjee, and R. J. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the IEEE International Conference on Data Mining* - ICDM '04, 2004, pp. 333-344.

[11] Bilenko, M., Basu, S., and Mooney, R. J. 2004. Integrating constraints and metric learning in semi-supervised clustering. In *Proceedings of the Twenty-First international Conference on Machine Learning* (Banff, Alberta, Canada, July 04 - 08, 2004). ICML '04, vol. 69. ACM, New York, NY, pp. 11. DOI= http://doi.acm.org/10.1145/1015330.1015360.

[12] Blei, D. M., Ng, A. Y., and Jordan, M. I. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research, vol. 3* (Mar. 2003), pp. 993-1022. DOI= http://dx.doi.org/10.1162/jmlr.2003.3.4-5.993.

[13] Christina Christakou, Leonidas Lefakis, Spyros Vrettos, Andreas Stafylopatis, A Movie Recommender System Based on Semi-supervised Clustering, In *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce* (CIMCA-IAWTIC'05), vol.2, pp. 897-903, 2005.

[14] Das, A. S., Datar, M., Garg, A., and Rajaram, S. 2007. Google news personalization: scalable online collaborative filtering. In *Proceedings of the 16th international Conference on World Wide Web* (Banff, Alberta, Canada, May 08 - 12, 2007). WWW '07. ACM, New York, NY, pp. 271-280. DOI= http://doi.acm.org/10.1145/1242572.1242610.

[15] Deshpande, M. and Karypis, G. 2004. Item-based top-*N* recommendation algorithms. *ACM Trans. Inf. Syst.* 22, 1 (Jan. 2004), pp. 143-177. DOI= http://doi.acm.org/10.1145/963770.963776.

[16] Harpale, A. S. and Yang, Y. 2008. Personalized active learning for collaborative filtering. In *Proceedings of the 31st Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (Singapore, Singapore, July 20 - 24, 2008). SIGIR '08. ACM, New York, NY, pp. 91-98. DOI= http://doi.acm.org/10.1145/1390334.1390352.

[17] T. Hofmann. Probabilistic latent semantic analysis. In *UAI '99: Proceedings of 15th Conference on Uncertainty in Artificial Intelligence*, pp. 289-296, 1999.

[18] Jin, R. and Si, L. 2004. A Bayesian approach toward active learning for collaborative filtering. In *Proceedings of the 20th Conference on Uncertainty in Artificial intelligence* (Banff, Canada, July 07 - 11, 2004). ACM International Conference Proceeding Series, vol. 70. AUAI Press, Arlington, Virginia, pp. 278-285.

[19] Popescul, A., Ungar, L. H., Pennock, D. M., and Lawrence, S. 2001. Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments. In *Proceedings of the 17th Conference in Uncertainty in Artificial intelligence* (August 02 - 05, 2001). J. S. Breese and D. Koller, Eds. Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers, San Francisco, CA, pp. 437-444.

[20] Luo Si and Rong Jin. (2003). Flexible Mixture Model for Collaborative Filtering, In *Proceedings of the Twentieth International Conference on Machine Learning* – ICML'03. Washington, DC USA, pp. 704-711.

[21] Takács, G., Pilászy, I., Németh, B., and Tikk, D. 2008. Matrix factorization and neighbor based algorithms for the netflix prize problem. In *Proceedings of the 2008 ACM Conference on Recommender Systems* (Lausanne, Switzerland, October 23 - 25, 2008). RecSys '08. ACM, New York, NY, pp. 267-274. DOI= http://doi.acm.org/10.1145/1454008.1454049.

[22] Mylonas, P., D. Vallet, P. Castells, M. Fernández And Y. Avrithis (2008). Personalized information retrieval based on context and ontological knowledge. *The Knowledge Engineering Review*, vol. 23, pp. 73-100 DOI=10.1017/S0269888907001282.

[23] Peng, W. 2009. Equivalence between nonnegative tensor factorization and tensorial probabilistic latent semantic analysis. In *Proceedings of the 32nd international ACM SIGIR Conference on Research and Development in information Retrieval* (Boston, MA, USA, July 19 - 23, 2009). SIGIR '09. ACM, New York, NY, pp. 668-669. DOI= http://doi.acm.org/10.1145/1571941.1572069.

[24] Wetzker, R., Umbrath, W., and Said, A. 2009. A hybrid approach to item recommendation in folksonomies. In *Proceedings of the WSDM '09 Workshop on Exploiting Semantic Annotations in information Retrieval* (Barcelona, Spain, February 09 - 09, 2009). ESAIR '09. ACM, New York, NY, pp. 25-29. DOI= http://doi.acm.org/10.1145/1506250.1506255.

[25] A. Said, R. Wetzker, W. Umbrath, L. Hennig, A hybrid PLSA approach for warmer cold start in folksonomy recommendation, In *Proceedings of the RecSys'09 Workshop on Recommender Systems & The Social Web*, New York, NY pp. 87-90

[26] Savia, E., Puolamäki, K., and Kaski, S. 2009. Latent grouping models for user preference prediction. Journal of *Machine Learning* vol. 74, issue 1 (Jan. 2009), pp. 75-109. DOI= http://dx.doi.org/10.1007/s10994-008-5081-7.

[27] Ma, H., King, I., and Lyu, M. R. 2009. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR Conference on Research and Development in information Retrieval* (Boston, MA, USA, July 19 - 23, 2009). SIGIR '09. ACM, New York, NY, pp. 203-210. DOI= http://doi.acm.org/10.1145/1571941.1571978.

[28] Ma, H., Yang, H., Lyu, M. R., and King, I. 2008. SoRec: social recommendation using probabilistic matrix factorization. In *Proceeding of the 17th ACM Conference on information and Knowledge Management* (Napa Valley, California, USA, October 26 - 30, 2008). CIKM '08. ACM, New York, NY, pp. 931-940. DOI= http://doi.acm.org/10.1145/1458082.1458205.

[29] Mei, Q., Cai, D., Zhang, D., and Zhai, C. 2008. Topic modeling with network regularization. In *Proceeding of the 17th international Conference on World Wide Web* (Beijing, China, April 21 - 25, 2008). WWW '08. ACM, New York, NY, pp. 101-110. DOI= http://doi.acm.org/10.1145/1367497.1367512.

[30] Yang, Y. and Hu, B. 2007. Pairwise Constraints-Guided Non-negative Matrix Factorization for Document Clustering. In *Proceedings of the IEEE/WIC/ACM international Conference on Web intelligence* (November 02 - 05, 2007). Web Intelligence. IEEE Computer Society, Washington, DC, pp. 250-256. DOI= http://dx.doi.org/10.1109/WI.2007.84.

[31] Xue, G., Dai, W., Yang, Q., and Yu, Y. 2008. Topic-bridged PLSA for cross-domain text classification. In *Proceedings of the 31st Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (Singapore, Singapore, July 20 - 24, 2008). SIGIR '08. ACM, New York, NY, pp. 627-634. DOI= http://doi.acm.org/10.1145/1390334.1390441.

[32] Hong C, Chen W, Zheng W, Shan J, Chen Y, Zhang Y, Parallelization and Characterization of Probabilistic Latent Semantic Analysis. *37th International Conference on Parallel Processing* – ICPP'08, 2008, pp.628-635.

[33] Hofmann, T. 1999. Probabilistic latent semantic indexing. In *Proceedings of the 22nd Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (Berkeley, California, United States, August 15 - 19, 1999). SIGIR '99. ACM, New York, NY, pp. 50-57. DOI= http://doi.acm.org/10.1145/312624.312649.

[34] Cai, D., Wang, X., and He, X. 2009. Probabilistic dyadic data analysis with local and global consistency. In *Proceedings of the 26th Annual international Conference on Machine Learning* (Montreal, Quebec, Canada, June 14 - 18, 2009). ICML '09, vol. 382. ACM, New York, NY, pp. 105-112. DOI= http://doi.acm.org/10.1145/1553374.1553388.

[35] Golbeck, J. A. 2005 *Computing and Applying Trust in Web-Based Social Networks*. Doctoral Thesis. UMI Order Number: AAI3178583., University of Maryland at College Park.

[36] Basu, S., Bilenko, M., and Mooney, R. J. 2004. A probabilistic framework for semi-supervised clustering. In *Proceedings of the Tenth ACM SIGKDD international Conference on Knowledge Discovery and Data Mining* (Seattle, WA, USA, August 22 - 25, 2004). KDD '04. ACM, New York, NY, pp. 59-68. DOI= http://doi.acm.org/10.1145/1014052.1014062.

[37] Marlin B., Modeling User Rating Profiles For Collaborative Filtering. In *Proceedings of 17th Neuro-Information Processing Systems conference, NIPS, 2003,* pp. 627-634

[38] Marlin B. , Collaborative Filtering from a ML Perspective, *Master's thesis, University of Toronto,* 2004.

[39] Monay, F. and Gatica-Perez, D. 2004. PLSA-based image auto-annotation: constraining the latent space. In *Proceedings of the 12th Annual ACM international Conference on Multimedia* (New York, NY, USA, October 10 - 16, 2004). MULTIMEDIA '04. ACM, New York, NY, pp. 348-351. DOI= http://doi.acm.org/10.1145/1027527.1027608.

[40] Kato, M., Kosaka, T., Ito, A., Makino, S., Fast and Robust Training of a Probabilistic Latent Semantic Analysis Model by the Parallel Learning and Data Segmentation. *Journal of Communication and Computer,* V6, N5, pp. 28-35, 2009.

[41] Belkin, M., Niyogi, P., and Sindhwani, V. 2006. Manifold Regularization: A Geometric Framework for Learning from Labeled and Unlabeled Examples. J*ournal Machine Learning Research* vol. 7 (Dec. 2006), pp. 2399-2434.

[42] Plangprasopchok A. and Lerman K., Exploiting social annotation for automatic resource discovery. In *AAAI workshop on Information Integration from the Web*, 2007, pp. 86-91

[43] Pitsilis, G., Knapskog, S., Social Trust as a solution to address sparsity-inherent problems of Recommender systems. *ACM RecSys 2009 Workshop on Recommender Systems & The Social Web*. Oct, 2009. pp. 33-40

[44] Kagie, M., van der Loos, M., and van Wezel, M. 2009. Including item characteristics in the probabilistic latent semantic analysis model for collaborative filtering. *AI Communications* 22, 4 (Dec. 2009), pp. 249-265.

[45] J. Gao, P.-N. Tan, and H. Cheng. Semi-supervised clustering with partial background information. In *Proceedings of the Sixth SIAM International Conference on Data Mining*, 2006, pp. 487-491.

[46] Wagstaff, K., Cardie, C., Rogers, S., and Schrödl, S. 2001. Constrained K-means Clustering with Background Knowledge. In *Proceedings of the Eighteenth international Conference on Machine Learning* (June 28 - July 01, 2001). C. E. Brodley and A. P. Danyluk, Eds. Morgan Kaufmann Publishers, San Francisco, CA, pp. 577-584.

[47] J. Zhu. 2005. Semi-supervised learning literature survey. *Computer Sciences Technical Report TR 1530*, University of Wisconsin-Madison. Available at http://www.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf, Last visited on: 17.05.2010

[48] Li, Q. and Kim, B. M. 2003. Clustering Approach for Hybrid Recommender System. In *Proceedings of the 2003 IEEE/WIC international Conference on Web intelligence* (October 13 - 17, 2003). Web Intelligence. IEEE Computer Society, Washington, DC, pp. 33.

[49] Givon, S. and Lavrenko, V. 2009. Predicting social-tags for cold start book recommendations. In *Proceedings of the Third ACM Conference on Recommender Systems* (New York, New York, USA, October 23 - 25, 2009). RecSys '09. ACM, New York, NY, pp. 333-336. DOI= http://doi.acm.org/10.1145/1639714.1639781.

[50] Moghaddam, S., Jamali, M., Ester, M., and Habibi, J. 2009. FeedbackTrust: using feedback effects in trust-based recommendation systems. In *Proceedings of the Third ACM Conference on Recommender Systems* (New York, New York, USA, October 23 - 25, 2009). RecSys '09. ACM, New York, NY, pp. 269-272. DOI= http://doi.acm.org/10.1145/1639714.1639765.

[51] A. Krithara, C. Goutte, J.M. Renders, and M.R. Amini. Reducing the annotation burden in text classification. In *Proceedings of the 1st International Conference on Multidisciplinary Information Sciences and Technologies* (InSciT), Merida, Spain, 2006, pp. 25-28.

[52] Krithara, A., Amini, M. R., Renders, J., and Goutte, C. 2008. Semi-supervised document classification with a mislabeling error model. In *Proceedings of the IR Research, 30th European Conference on Advances in information Retrieval* (Glasgow, UK, March 30 - April 03, 2008). C. Macdonald, I. Ounis, V. Plachouras, I. Ruthven, and R. W. White, Eds. Lecture Notes In Computer Science. Springer-Verlag, Berlin, Heidelberg, pp. 370-381.

[53] Marlin B., Modeling user rating profiles for collaborative filtering. In *Advances in Neural Information Processing Systems- NIPS'04*, volume 16, 2004

[54] Hofmann, T., Puzicha J,, Jordan M., Learning from dyadic data, *Proceedings of the 1998 conference on Advances in neural information processing systems* II, p.466-472, July 1999

[55] Xue, G., Lin, C., Yang, Q., Xi, W., Zeng, H., Yu, Y., and Chen, Z. 2005. Scalable collaborative filtering using cluster-based smoothing. In *Proceedings of the 28th Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (Salvador, Brazil, August 15 - 19, 2005). SIGIR '05. ACM, New York, NY, pp. 114-121. DOI= http://doi.acm.org/10.1145/1076034.1076056

[56] Friend of a friend project website, http://www.foaf-project.org/, last visted on 10.01.2010.

[57] Netflix competition web page, http://www.netflixprize.com/, last visited on 20.01.2010.

[58] Association for Computing Machinery, recommender systems conference web page, http://recsys.acm.org/, Last visited on 11.01.2010.

[59] Netflix prize winner team Bellkor's Pragmatic Chaos web site, http://www2.research.att.com/~volinsky/netflix/bpc.html, Last visited on 20.01.2010.

[60] Melville, P., Mooney, R. J., and Nagarajan, R. 2002. Content-boosted collaborative filtering for improved recommendations. In *Eighteenth National Conference on Artificial intelligence* (Edmonton, Alberta, Canada, July 28 - August 01, 2002). Dechter, R., Kearns, M., and Sutton, R., Eds. American Association for Artificial Intelligence, Menlo Park, CA, pp. 187-192.

[61] Truyen, T. T., Phung, D. Q., and Venkatesh, S. 2007. Preference networks: probabilistic models for recommendation systems. In *Proceedings of the Sixth Australasian Conference on Data Mining and Analytics - Volume 70* (Gold Coast, Australia, December 03 - 04, 2007). P. Christen, P. Kennedy, J. Li, I. Kolyshkina, and G. Williams, Eds. ACM International Conference Proceeding Series, vol. 311. Australian Computer Society, Darlinghurst, Australia, pp. 195-202.

[62] Popescul, A., Ungar, L. H., Pennock, D. M., and Lawrence, S. 2001. Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-

Data Environments. In *Proceedings of the 17th Conference in Uncertainty in Artificial intelligence* (August 02 - 05, 2001). J. S. Breese and D. Koller, Eds. Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers, San Francisco, CA, pp. 437-444.

[63] Harpale, A. S. and Yang, Y. 2008. Personalized active learning for collaborative filtering. In *Proceedings of the 31st Annual international ACM SIGIR Conference on Research and Development in information Retrieval* (Singapore, Singapore, July 20 - 24, 2008). SIGIR '08. ACM, New York, NY, pp. 91-98. DOI= http://doi.acm.org/10.1145/1390334.1390352.

[64] Shashanka M., Raj B., and Smaragdis P., Probabilistic latent variable models as nonnegative factorizations. *Computational intelligence and neuroscience*, 2008, Published online, doi: 10.1155/2008/947438 .

[65] Koren Y., Bell R., Volinsky C., "Matrix Factorization Techniques for recommender Systems," IEEE *Computer Journal*, vol. 42, no. 8, pp. 30-37, Aug. 2009, doi:10.1109/MC.2009.263.

[66] Burke, R. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction* 12, 4 (Nov. 2002), pp. 331-370. DOI= http://dx.doi.org/10.1023/A:1021240730564.

[67] Bakshy, E., Karrer, B., and Adamic, L. A. 2009. Social influence and the diffusion of user-created content. In *Proceedings of the Tenth ACM Conference on Electronic Commerce* (Stanford, California, USA, July 06 - 10, 2009). EC '09. ACM, New York, NY, pp. 325-334, DOI= http://doi.acm.org/10.1145/1566374.1566421

[68] Hofmann, T. 2003. Collaborative filtering via Gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th Annual international ACM SIGIR Conference on Research and Development in informaion Retrieval* (Toronto, Canada, July 28 - August 01, 2003). SIGIR '03. ACM, New York, NY, pp. 259-266. DOI= http://doi.acm.org/10.1145/860435.860483.

[69] Deerwester, S., Dumais, S. T., Furnas, G. W., Thomas, Harshm*an, R., Indexing by latent semantic analysis, Journal of the American Society for Information Science*, Vol. 41, pp. 391-407, 1990.

[70] CiteULike website, http://www.citeulike.org/ , Last visited on 21.01.2010.

[71] Orkut social networking website, http://www.orkut.com/, Last visited on 21.01.2010.

[72] Bibsonomy website, http://www.bibsonomy.org/, Last visited on 21.01.2010.

[73] Movielens dataset website, http://www.grouplens.org/taxonomy/term, Last visited on 21.01.2010.

[74] D. Blei and J. Lafferty. Topic Models. In A. Srivastava and M. Sahami, editors, *Text Mining: Theory and Applications*. Taylor and Francis, 2009.

[75] Blei, D. M., Ng, A. Y., and Jordan, M. I. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* 3 (Mar. 2003), 993-1022. DOI= http://dx.doi.org/10.1162/jmlr.2003.3.4-5.993

[76] Rish I., An empirical study of the naive bayes classifier. In *Proceedings of IJCAI-01 workshop on Empirical Methods in AI*, pp. 41--46, Sicily, Italy, 2001.

[77] Kullback, S. and Leibler, A., On information and sufficiency. *Annals of Mathematical Statistics*, vol. 22 pp. 79--86, 1951.

[78] Salton, G., Wong, A., and Yang, C. S. 1975. A vector space model for automatic indexing. *Commuicaction of the ACM* vol. 18, issue 11 (Nov. 1975), 613-620. DOI= http://doi.acm.org/10.1145/361219.361220.

[79] Cai, D., Mei, Q., Han, J., and Zhai, C. 2008. Modeling hidden topics on document manifold. In *Proceeding of the 17th ACM Conference on information and Knowledge Management* (Napa Valley, California, USA, October 26 - 30, 2008).

CIKM '08. ACM, New York, NY, 911-920. DOI= http://doi.acm.org/10.1145/1458082.1458202.

[80] Adomavicius,G., Tuzhilin, A., Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, *IEEE Transactions on Knowledge and Data Engineering*, v.17 n.6, p.734-749, June 2005.

[81] Pilászy, I. and Tikk, D. 2009. Recommending new movies: even a few ratings are more valuable than metadata. In *Proceedings of the Third ACM Conference on Recommender Systems* (New York, New York, USA, October 23 - 25, 2009). RecSys '09. ACM, New York, NY, 93-100. DOI= http://doi.acm.org/10.1145/1639714.1639731.

[82] Victor, P., Cornelius, C. ,DeCock, M., Teredesai, A, A Comparative Analysis of Trust-Enhanced Recommenders for Controversial Items, *Third International Conference on Weblogs and Social Media*, ICWSM 2009.

[83] Moletrust example, http://www.trustlet.org/wiki/Moletrust, Last visited on 23.04.2010.

[84] Joung, J., Shin, D., Seong, R. H., and Zhang, B. 2006. Identification of regulatory modules by co-clustering latent variable models: stem cell differentiation. *Bioinformatics* 22, 16 (Aug. 2006), 2005-2011. DOI= http://dx.doi.org/10.1093/bioinformatics/btl343.

[85] Xiaoyuan Su and Taghi M. Khoshgoftaar, A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*, vol. 2009, Article ID 421425, 19 pages, 2009.

[86] J. Golbeck. Generating predictive movie recommendations from trust in social networks. In *Proceedings of The Fourth International Conference on Trust Management*, 2006.

[87] Paolo Massa , Paolo Avesani, Trust-aware recommender systems, *Proceedings of the 2007 ACM conference on Recommender systems*, October 19-20, 2007, Minneapolis, MN, USA, 17-24.

[88] Epinions product review website, http://www.epinions.com, Last visited on 05.05.2010.

[89] Netflix movie rental-recommendation website, http://www.netflix.com, Last visited on 05.05.2010.

[90] IMDB data source, http://www.imdb.com/interfaces#plain, Last visited on 05.05.2010.

[91] Marlin, B. M. and Zemel, R. S. 2009. Collaborative prediction and ranking with non-random missing data. In *Proceedings of the Third ACM Conference on Recommender Systems*, RecSys '09. ACM, New York, pp. 5-12.

[92] M. Brunato, R. Battiti, A. Villani, and A. Delai. A location-dependent recommender system for the web. *Technical report, DIT--02--0093, University of Trento, Dept of Information and Communication Technology*, November 2002.