CONTRIBUTIONS À LA DÉTECTION DES ANOMALIES ET AU DÉVELOPPEMENT DES SYSTÈMES DE RECOMMANDATION

par

Wu Shu

Thèse présentée au Département d'informatique
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)

FACULTÉ DES SCIENCES
UNIVERSITÉ DE SHERBROOKE

Sherbrooke, Québec, Canada, juillet 2012

Library and Archives
Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-93263-6

Our file Notre référence
ISBN: 978-0-494-93263-6

CONTRIBUTIONS TO OUTLIER DETECTION AND RECOMMENDATION SYSTEMS

Shu Wu
Faculté des Sciences
Université de Sherbrooke

A thesis submitted for the degree of
Doctor of Philosophy
July 2012

Le 20 juillet 2012

le jury a accepté la thèse de Monsieur Shu Wu
dans sa version finale.

Membres du jury

Professeur Shengrui Wang
Directeur de recherche
Département d'informatique

Professeur André Mayers
Évaluateur interne
Département d'informatique

Professeur Philippe Fournier-Viger
Évaluateur externe
Département d'informatique
Université de Moncton

Professeur Ernest Monga
Président rapporteur
Département de mathématiques

Acknowledgements

First and foremost, I would like to thank my supervisor Professor Shengrui Wang for his dedicated supervision during my Ph.D. study. I have benefited so much in my research from the inspiration, motivation and encouragement that Prof. Wang constantly provided. He introduced me to the research field of outlier detection and recommendation systems which really matched my skill-set and interest. I also like to thank him for giving me the abundant freedom to explore various research directions. Working with him was really memorable and fun experience.

I was partly supported by the Ph.D. Scholarship program of the China Scholarship Council (CSC) during my studies at the Université de Sherbrooke. Part of the research work reported in this thesis was funded by Discovery and Discovery Accelerator Supplements programs of Natural Sciences and Engineering Research Council of Canada (NSERC) granted to Prof. Shengrui Wang. I would like to express my sincere thanks to NSERC and CSC.

I would also like to thank my committee president Dr. Ernest Monga, committee members, Dr. André Mayers, Dr. Philippe Fournier-Viger, and Dr. Shengrui Wang for their time and their feedback on my thesis. Their reviews and comments are really helpful to improve the thesis.

I would like to thank all my colleagues and friends in Prospectus Lab and in Université de Sherbrooke for sharing the good time in Sherbrooke.

Finally, I would like to thank my parents and my brother for their endless encouragement and belief in me.

Sommaire

Le forage de données, appelé également "Découverte de connaissance dans les bases de données", est un jeune domaine de recherche interdisciplinaire. Le forage de données étudie les processus d'analyse de grands ensembles de données pour en extraire des connaissances, et les processus de transformation de ces connaissances en des structures faciles à comprendre et à utiliser par les humains. Cette thèse étudie deux tâches importantes dans le domaine du forage de données : la détection des anomalies et la recommandation de produits. La détection des anomalies est l'identification des données non conformes aux observations normales. La recommandation de produit est la prédiction du niveau d'intérêt d'un client pour des produits en se basant sur des données d'achats antérieurs et des données socio-économiques. Plus précisément, cette thèse porte sur 1) la détection des anomalies dans de grands ensembles de données de type catégorielles; et 2) les techniques de recommandation à partir des données de classements asymétriques.

La détection des anomalies dans des données catégorielles de grande échelle est un problème important qui est loin d'être résolu. Les méthodes existantes dans ce domaine souffrent d'une faible efficience et efficacité en raison de la dimensionnalité élevée des données, de la grande taille des bases de données, de la complexité élevée des tests statistiques, ainsi que des mesures de proximité non adéquates. Cette thèse propose une définition formelle d'anomalie dans les données catégorielles ainsi que deux algorithmes efficaces et efficients pour la détection des anomalies dans les données de grande taille. Ces algorithmes ont besoin d'un seul paramètre : le nombre des anomalies. Pour déterminer la valeur de ce paramètre, nous avons développé un critère en nous basant sur un nouveau concept qui est l'holo-entropie.

Plusieurs recherches antérieures sur les systèmes de recommandation ont négligé un type de classements répandu dans les applications Web, telles que le commerce électronique (ex. Amazon, Taobao) et les sites fournisseurs de contenu (ex. YouTube). Les données de classements recueillies par ces sites se différencient de celles de classements des films et des musiques par leur distribution asymétrique élevée. Cette thèse propose un cadre mieux adapté pour estimer les classements et les préférences quantitatives d'ordre supérieur pour des données de classements asymétriques. Ce cadre permet de créer de nouveaux modèles de recommandation en se basant sur la factorisation de matrice ou sur l'estimation de voisinage. Des résultats expérimentaux sur des ensembles de données asymétriques indiquent que les modèles créés avec ce cadre ont une meilleure performance que les modèles conventionnels non seulement pour la prédiction de classements, mais aussi pour la prédiction de la liste des *Top-N* produits.

Abstract

Data mining, also called Knowledge Discovery in Databases, is a relatively young and interdisciplinary research field of computer science. It is the process of analyzing large-scale datasets, extracting knowledge, and then transforming this knowledge into a human-understandable structure for further use. Outlier detection and recommendation systems are two important tasks in data mining. Outlier detection refers to detecting observations in a given dataset that do not conform to normal observations, while recommendation systems try to predict user's preference towards items from historic data of purchase and other related socio-economic data of the users. The main focus of this thesis is to study two key issues in outlier detection and recommendation systems: outlier detection from (or in) large-scale categorical datasets and recommendation systems from highly-skewed rating datasets.

Detecting outliers in large-scale categorical datasets is a very important and open significant topic in outlier detection. Existing methods in this area suffer from low effectiveness and low efficiency due to high dimensionality and large size of the datasets, high-complexity of statistical tests or inefficient proximity-based measures. In this thesis, we provide a formal definition of outlier in the categorical datasets, and design two effective and efficient algorithms with only one parameter for the task of outlier detection in large-scale categorical datasets.

Previous research on recommendation systems has neglected one significant rating scenario, which broadly exists in many real Web applications, such as e-commerce (e.g. Amazon, Taobao) and content provider websites (e.g. Youtube). The rating datasets collected from these websites have different characteristics from the traditional movie and music rating datasets. Their ratings distributions are with high skewness. After examining the properties of this kind of rating datasets, we propose a new framework for estimating rating and quantitative high-order preference for skewed rating datasets. This framework allows to generate novel and more effective matrix factorization and neighborhood models. Experimental results on typical highly-skewed datasets show that new models created under this framework can generate better performance than the conventional methods on the skewed rating datasets for not only rating prediction but also for *Top-N* recommendation.

Contents

List of Figures

Chapter 1

Introduction

Data mining is a relatively young research field of computer science. Utilizing methods at the intersection of artificial intelligence, machine learning, statistics, and database systems, data mining aims to extract knowledge from tremendous data and transform it into a human-understandable structure for further use. Outlier detection and recommendation systems are two fundamental tasks in this research area. In this chapter, we will review the background of these two tasks.

The structure of our introduction is as follows. Section 1 focuses on discussion of outlier detection, including outlier definitions, outlier detection's applications, classification of existing methods, and evaluation metrics. Section 2 describes the basic concepts about recommendation systems including classification of existing methods, background and classification of collaborative filtering, as well as evaluation metrics. Finally, we conclude this chapter with a discussion of the contributions of this thesis on outlier detection and recommendation systems, and provide the related publication list of the author. The materials in this chapter help to understand the subsequent chapters of this thesis.

1.1 Outlier Detection

The datasets collected from the real world always suffer from unusual observations [34]. These unusual objects may be "due to several factors, including: ignorance and human errors, rounding errors, transcription error, inherent variability of the domain, instrument malfunction and biases" [34]. These observations may affect the application of an advanced data analysis method, but may also indicate interesting phenomena or findings resulted from rare but correct actions/behaviour, and motivate further investigation.

Outlier detection is an important and challenging task that has been treated within diverse domains and research areas such as statistics, machine learning, data mining, information theory [22, 52, 23, 117]. Generally, in data mining, outlier detection refers to the problem of finding and, where appropriate, removing objects in a dataset which are considerably dissimilar, exceptional and inconsistent w.r.t. the majority of objects in a dataset [6]. These non-conforming objects are called outliers, also referred to as anomalies, surprises, aberrations, exceptions, surprises, novelties, peculiarities, contaminants, etc, in different domains [6, 22]. Correspondingly, the problem of identifying unusual observations is named as outlier detection, novelty detection, anomaly detection, noise detection, deviation detection or exception mining [22].

The term outlier originally stems from the field of statistics [52]. Previous work in statistics, machine learning and data mining, has proposed several definitions for an outlier, but seemingly there does not exist a universally accepted definition [6]. Here, we list some classical definitions of an outlier or outliers

Definition 1. (Hawkins' definition [45]) *An outlier is an observation, which deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism.*

Definition 2. (Grubbs' definition) [38] *An outlying observation, or outlier, is one that appears to deviate markedly from other members of the sample in which it occurs.*

Definition 3. (Definition of Barnett and Lewis) [123] *An outlier is an observation (or subset of observations) which appears to be inconsistent with the remainder of that set of data.*

Definition 4. (Definition of Moore and McCabe) [91] *An outlier is an observation that lies outside the overall pattern of a distribution.*

Definition 5. (Definition of Aggarwal and Yu) [6] *Outliers may be considered as noise points lying outside a set of defined clusters, or alternatively outliers may be defined as the points that lie outside of the set of clusters but are also separated from the noise.*

These definitions capture the meaning of outliers from a general point of view. There are many other definitions of outliers [64, 92, 49], which are dependent on particular detection methods.

Outlier detection not only can be implemented as a pre-processing step prior to the application of an advanced data analysis method, but also can be used as an effective tool to discover interest patterns such as the expense behavior of a to-be-bankrupt credit cardholder. The process of outlier detection is an essential step in a variety of practical applications including intrusion detection [71], health system monitoring [52] and criminal activity detection in E-commerce [8], and can also be used in scientific research for data analysis and knowledge discovery in biology, chemistry, astronomy, oceanography and other fields [52]. There are is some typical applications, e.g. fraud detection, intrusion detection, fault diagnosis, satellite image analysis, medical condition monitoring, public health monitoring, etc [52].

1.1.1 Approaches of Existing Methods

According to [22, 52], if the existing methods for outlier detection are classified according to the availability of labels in the training datasets, there are three fundamental categories: supervised, semi-supervised and unsupervised approaches. The general idea of three broad categories of outlier detection techniques are discussed below.

The supervised approach makes an assumption that the domain knowledge on both normal and abnormal data exists and can be used to build a classification model. This approach learns classifier from the labelled objects and assigns appropriate labels to test objects. If a test object lies in a region of normality it is classified as normal, otherwise it is flagged as an outlier. Sometimes, this classification problem may be highly imbalanced, and may contain multiple normal and/or abnormal classes. The supervised approach to outlier detection has been studied extensively and many methods have been developed [39, 13, 119, 108, 35, 57].

The semi-supervised approach [134, 40] constructs a model representing normal behavior from a given training dataset of normal objects, and then computes the likelihood of a test object's being generated by this model. This semi-supervised approach is more applicable than the previous approach since only labelled normal objects are required. However, this approach tends to classify previously unseen normal objects as outliers, causing high false alarm rate.

Requiring no prior knowledge of the dataset, the unsupervised outlier detection approach detecting outliers in an unlabeled dataset [10, 6, 110, 127, 76] is based on the assumption that the majority of objects in this dataset are normal. This approach is more widely applicable and popular, as in most applications there are no training data available. The remainder of this section is devoted for the classification of the unsupervised approach, since the unsupervised scenario is our focus in this thesis.

1.1.2 Classification of Unsupervised Outlier Detection Methods

Unsupervised approach to outlier detection encompasses a broad spectrum of techniques, drawn from the full gamut of computer science and statistics. The existing detection methods in this approach primarily can be classified into four groups: statistics-based methods, clustering-based methods, distance-based methods and density-based methods [43]. Since Chapter 2 provides detailed review on the unsupervised methods of categorical datasets, here we focus on the important methods of numerical datasets. The detailed introduction about these four groups is given as follows.

In a typical statistics-based method, the normal objects are assumed to follow a known distribution, e.g. Gaussian, Poisson, etc., and outliers deviate strongly from this distribution. If the underlying distribution is not known, a searching process is required to find out the best distribution to fit with the dataset. But this process is very time consuming and does not always work, especially for data that come from different sources with different distributions. Furthermore, for many applications, the underlying distribution is unknown [124]. Overall, statistics-based techniques are simple in principles, but inapplicable for data with more than three dimensions. Related work about statistics-based methods can be found in [13].

Clustering-based methods are based on the assumption that normal data points belong to large and dense clusters while outliers do not. The typical framework of such methods can be described as: performing a clustering process on the dataset; analyzing the obtained clusters to assess their significance; outputting outliers which are objects that do not fit into any clusters or belong to clusters with low membership. Usually outliers lack formal definition in the clustering-based approach, and are by-products of clustering. This limits capabilities of clustering-based method in providing intuition on the detected results. Some examples of this category can be found in [39, 40, 6, 47].

Distance-based outlier detection methods generally exploit distances of objects to their corresponding neighborhood in a dataset. These methods use a candidate's average distance to its k nearest neighbours [11] (or alternatively, the distance to its kth nearest neighbour [98]) as the anomalous score and return the top few objects in a dataset whose score is the highest. These methods also can simply count the total number r-neighbours, i.e. the number of data points within the distance r, of each object [64]. Normally, distance-based methods do not assume any distribution of the dataset as statistical techniques do, but suffer expensive computational cost of searching nearest neighbourhood.

Density-based methods, e.g. LOF [17], LOCI [93], generally assign to each object a factor describing the relative density of this object's neighbourhood. Similar to distance-based approach, density-based approach also involves in the computation of objects' nearest neighbours. However, the measurement of an object to its nearest neighbours is then compared to the same measurements of neighbours. The purpose of doing so is to overcome different effects of dense and sparse clusters on points' neighbourhood in detecting outliers. Computational costs of these methods become even more expensive than that of distance-based methods. Because of the applicability for large and high-dimensional data, such kind of methods still attract much attention from the research community [75, 17, 93].

1.2 Recommendation Systems

Recommendation systems (or recommender systems) try to profile a user preference over items by the user feedback and seek to recommend items from the overwhelming set of choices to fit user's tastes. A more specific definition of recommendation systems is given by Burke [18].

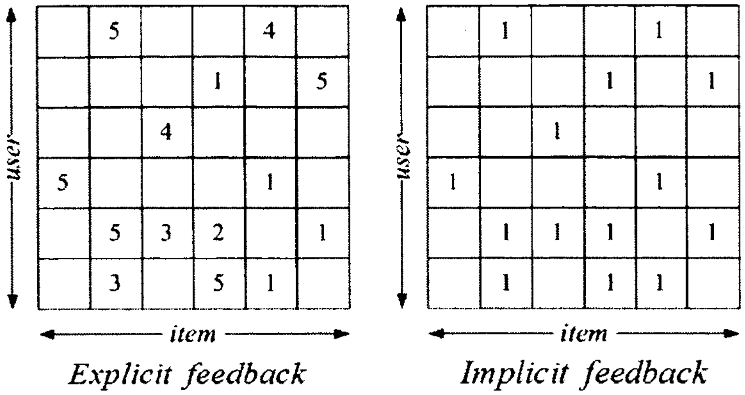
Definition 6. (Burke's definition) *Systems that produce individualized recommendations as output or have the effect of guiding the user in a personalized way to interesting or useful objects in a large space of possible options.*

Individualized (or personalized) is the main keyword in this formal definition. This term indicates that each user will be provided with different items by the recommendation systems.

Recently, recommendation systems have become more and more popular on application websites. For instance, web-based services help users in discovering interesting products on Amazon[1], promising movies on Netflix[2], videos on Youtube[3] and Hulu[4], websites on StumbleUpon[5], news on Digg[6], music on Last.fm[7] and iTunes[8], and social content and users on Facebook[9], Twitter[10], etc. These recommendation systems can generate personalized recommended items which well match users' taste. According to [70], two thirds of the movies rented by Netflix are recommended, Google news recommendations result in 38% more clickthroughs, and 35% of the product sales on Amazon.com are recommended items.

In recommendation systems, there are two types of user feedback, i.e. explicit feedback and implicit feedback, which are utilized in profiling the preference of users. These two kinds of feedbacks are illustrated in the user-item matrixes in Fig. 1.1. In

[1] http://www.amazon.com/
[2] http://www.netflix.com/
[3] http://www.youtube.com/
[4] http://www.hulu.com/
[5] http://www.stumbleupon.com/
[6] http://www.digg.com/
[7] http://www.last.fm/
[8] http://www.apple.com/itunes/
[9] http://www.facebook.com/
[10] http://twitter.com/



	5			4	
			1		5
		4			
5				1	
	5	3	2		1
	3		5	1	

Explicit feedback

	1			1	
			1		1
		1			
1				1	
	1	1	1		1
	1		1	1	

Implicit feedback

Figure 1.1: Examples of two different user feedbacks illustrated in the user-item matrixes. Explicit user feedback is listed in the left matrix, and implicit feedback is demonstrated in the right matrix.

the case of explicit feedback, users explicitly express their opinion by rating values towards items. The rating indicates how a user feels about a particular item. In contrast, implicit feedback is inferred from observing user behaviors. The implicit feedback of a particular user is generated from the watching/browsing/purchasing actions of this user. For instance, a user listens to a song for a long time, from which we can infer that the user like this song. Implicit feedback can be collected from various sources, such as number of times used, web click-through, purchase action, etc. Normally, a distinction between explicit and implicit feedback needs to be made for building a recommendation system. Our research work in this thesis deals with the recommendation problem with the explicit feedback.

For recommendation from the explicit feedback, rating prediction is the typical and concrete task, where the objective of this task is to predict the 'rating' that a user would give to an item (such as music, books, or movies) or social element (e.g. people or groups) they had not yet considered, using a model built from the characteristics of items (content-based approaches) or the user's social environment (collaborative filtering approaches) [102].

Previously proposed methods for building recommendation systems can be categorized into three primary approaches [4], respectively, Collaborative Filtering (CF), Content-Based filtering (CB) and a hybrid approach. Here, we summarize the basics concepts, and then respectively discuss the detailed information of these approaches.

1. Collaborative filtering collects and merges preference information of users, and generates predictions for an individual user based on similarity measurements of users and (or) items [30, 107, 53, 104].

2. Content-based filtering [36, 90, 95, 80] generates recommendations utilizing content profiles of items and profiles of users that describe the types of item the users like. In other words, this approach tries to recommend items which are similar to those that a user liked in the past.

3. Hybrid approach [19, 18] typically combines collaborative filtering and content-based filtering, and can be more effective in some cases.

1.2.1 Collaborative Filtering

Collaborative Filtering (CF) is the most popular and, to date, the most successful approach to recommendation systems.· CF collects and merges a large amount of users' rating information to predict what users will like based on similarity measurements among users and (or) among items [107, 53, 104]. The term collaborative filtering is first used in [41], which presents the Tapestry system to filter emails using collaborative filtering. Other important early work was done by [101] on their `Grouplens` system for recommending Usenet articles, and by [109] on their `Ringo` music recommender system.

Collaborative filtering methods can be categorized into two primary types according to [16], which are memory-based approach and model-based approach. Memory-based approach operates on an entire rating dataset to generate recommended items to a particular user, while model-based approach first manipulates the given ratings to build a model, which then can be used to predict rating values for a given user-item pair.

Memory-based methods

Memory-based methods utilize the entire user-item ratings to generate a prediction. In the training phase of a memory-based method, all ratings should be scanned and stored into the memory. Memory-based methods can be further divided into user-based and item-based methods, which are based on the K-Nearest Neighbour algorithm. The user-based method computes a set of K nearest neighbours of the target user by calculating the similarities between users' rating profiles. Once the neighbours are obtained, we can calculate the prediction rating value of the target user using a weighted average of the neighbours' item ratings. On the other hand, item-based method focuses on finding K similar items rather than similar users [107]. Correspondingly, for a target item, prediction can be generated by taking a weighted average of the target user's item ratings on these neighbour items. There are a variety of different ways to calculate the similarity between items or users. In a typical memory-based method, the most commonly-used similarity between users or similarity between items are calculated using cosine-based similarity [16] or Pearson correlation similarity [101].

Here, we would like to introduce the item-based method with cosine-based similarity [107], which will be used as an original model for generating a new and more effective model by our proposed RP framework in Chapter 3. At first, let us assume a set $\mathcal{U}$ of n users and a set $\mathcal{I}$ of m items in a typical CF scenario. Each user u is associated with a set $\mathcal{I}_u$, which contains all the items the user has rated. The dataset containing all users and all rated items is denoted as $\mathcal{D}_t \subset \mathcal{U} \times \mathcal{I}$. All observed ratings r_{ui} on the dataset $\mathcal{D}_t$ are denoted as the rating dataset $\mathcal{R}_t := \{r_{ui} | (u,i) \in \mathcal{D}_t\}$.

At the beginning, we need to compute the similarity matrix S which measures the similarities between the items in the set $\mathcal{I}$, where s_{ij} denotes the similarity of item i and item j. Here, we use the cosine-based similarity [16]. More concretely, the cosine-based similarity s_{ij} represents the cosine of the angle between two item vectors $\vec{i}$ and $\vec{j}$ in a n-dimension user-space. It is calculated by

$$s_{ij} = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{||\vec{i}||_2 * ||\vec{j}||_2}$$

where $\|\|\|_2$ means the L^2 norm of the vector.

After the computation of similarity matrix, we seek to calculate the prediction rating for an item based on a set $\mathcal{I}_u^k$ containing the k most similar items that the target user has rated. This prediction is generated by computing a weighted average of the user's ratings on these similar items.

$$\hat{r}_{ui} = \frac{1}{\sum_{m \in \mathcal{I}_u^k} |s_{im}|} \sum_{m \in \mathcal{I}_u^k} s_{im} \cdot r_{um}$$

Model-based methods

Model-based methods first learn a model of user behavior from a rating dataset in advance, and then use this model to generate recommendations. Compared with memory-based methods, model-based methods usually scale better in terms of their resource requirements (memory and computing time) and do not require keeping actual user profiles in memory for prediction. Besides, in a lot of applications, model-based methods outperform memory-based methods in terms of prediction accuracy [102].

Many different machine learning techniques have been used in the process of model building, such as Naive Bayes [16], restricted Boltzmann machines [105], graph-theoretic approach [5], and latent factor [53, 65, 66, 96]. Latent factor techniques have been generating much interest and progress recently, because of its attractive accuracy and scalability. These models reduce the dimensionality of the space of user-item ratings and try to map both items and users to a joint latent semantic space [65]. The rating value can be predicted by the inner products of a user and an item in this space. Examples of latent factor techniques applied to recommendation include such as Singular Value Decomposition (SVD) [26], Probabilistic Latent Semantic Analysis (PLSA) [53], and Matrix Factorization (MF) [116].

The MF method [116] is a simple and effective latent factor model. Here, we give a detailed introduction to this model, as it will be used as a competitor in Chapter 3. In this MF method, the f-dimension factor vectors $p_u \in \mathbb{R}^f$ and $q_i \in \mathbb{R}^f$ describe the latent characteristics of user u and item i, and the predicted rating of this user-item pair can be calculated by $\hat{r}_{ui} = q_i^T p_u$.

In order to estimate the latent vectors p_u and q_i, we can solve the following least squares problem

$$\sum\nolimits_{\mathcal{R}_t}(r_{ui}-q_i^Tp_u)^2+\lambda\left(\sum\nolimits_{|\mathcal{U}|}p_u^2+\sum\nolimits_{|\mathcal{I}|}q_i^2\right)$$

where r_{ui} means the given rating of user u to item i and λ controls the extent of regularization, which is usually estimated by cross validation. Model parameters are determined by minimizing this regularized squared error function through stochastic gradient descent [66]. Looping over all known ratings in $\mathcal{R}_t$, the updating function of parameters can be computed as follows

$$\begin{aligned} q_i &\leftarrow q_i+\gamma\left((r_{ui}-q_i^Tp_u)p_u^T+\lambda q_i\right)\\ p_u &\leftarrow p_u+\gamma\left((r_{ui}-q_i^Tp_u)q_i^T+\lambda p_u\right)\end{aligned}$$

where γ works as learning rates of these updating steps.

Advantages of collaborative filtering

Besides the effectiveness and scalability, the collaborative filtering approach has several other significant strengths.

1. The greatest strength of CF is that it does not require any content information about the product for recommendation. Thus, this approach is suitable to be implemented for complex items, such as music and movies, of which the content properties are difficult to extract [18].

2. Furthermore, the CF approach has the ability to recommend serendipitous items, which have very different content from the items that the user has chosen, and which the user would like but have not discovered yet [51].

3. Finally, according to [50], the CF approach takes into account the quality of items in recommendation, especially in the case of explicit feedback, and can prevent poor recommendations. For instance, two movies with same characteristic features have very different qualities, CF may find out the difference and

recommend the item with high quality.

Existing problems in collaborative filtering

According to the work presented in [55], the cold-start is one of the most serious problems of the CF approach. This problem refers to the situation where a recommendation system is in the start-up phase, or when a new user or item is added into the system. In this situation, the CF system has difficulty in generating recommendations.

The collaborative filtering approach has difficulty in predicting ratings from the sparse rating dataset, where some users have small sets of rated items [55]. According to Breese et al. [16], CF works well for a user only if a reasonable amount of ratings of this user is available.

Finally, in [55] the idea of non-transitive associations among users or items is presented. This means that if two similar items have never been rated by the same user, or if two similar users have never rated the same item, their relationship may be lost. In this case, CF will not treat those two items or two users as similar ones, and this may affect the performance of the system.

1.2.2 Content-based Filtering

Typically, Content-Based filtering (CB) [36, 90, 95, 80] creates a representative profile of a user's interest utilizing characteristic features of his/her rated items, and then recommend other unrated items likely being most relevant to that user. In other words, methods in this approach generate a weighted content-based profile for a user, where the values of this user's profile indicate the importance of corresponding features to this user, and then seek to recommend items which well suit the profile of this user. Using various techniques [36, 90, 95, 80], this weighted vector describing user's preference can be calculated from individual feature vectors of rated items.

The work [36] first presents a content-based information filtering, matching user interests to text documents using two matching methods and two types of user profiles. `Libra` system is a book recommender using Bayesian learning algorithm and extracts information of books for text categorization [90]. [95, 80] survey the field of content-based recommendation, including a method for representing items and user profiles, and a method for comparing items to the user to determine which to recommend.

The recommender system implemented on Pandora Internet Radio [11] is a popular example of CB. This system takes the features of an initial seed provided by a user to build a station, which plays music with similar properties to the user. Then the user's feedback on these played songs is used to learn the interest profile of this user. When the user likes a particular song, the system then emphasizes some certain features of this user, while this user dislikes a song, the system deemphasizes certain features. Other examples of content-based recommender systems include Rotten Tomatoes[12], Internet Movie Database[13], Jinni[14] and Rovi Corporation[15].

In contrast to CF, CB does not have cold-start problem for new items, since the features of a new item can be extracted when the item is added. In addition, the recommended items are more explainable than CF as they match the feature vector of user interests. The main disadvantage of CB is that it is difficult to extract good feature vectors of complex items with tremendous properties such as music and movies. If possible, creating feature vectors for these items is generally a very laborious process. In addition, the content-based filtering can only recommend items from a narrow topic range; they are unable to provide serendipitous recommendations [51].

1.2.3 Hybrid Approach

Hybrid recommendation systems were developed in the recent years as an attempt to overcome the weakness of pure content-based filtering or pure collaborative filtering methods. As stated in [18], "hybrid recommender systems combine two or more recommendation techniques to gain better performance with fewer of the drawbacks of any individual one." For example, this approach can be used to alleviate some

[11] http://www.pandora.com/

[12] http://www.rottentomatoes.com/

[13] http://www.imdb.com/

[14] http://www.jinni.com/

[15] http://www.rovicorp.com/

common problems, such as cold-start and sparsity, of other approaches [55]. The recommendation system of Netflix is a representative example of this approach, which offers movies recommendations based on users' previous ratings (using collaborative filtering), and the characteristics of watched movies (utilizing content-based filtering).

The hybrid approach can be implemented in many ways [4], for example by adding content-based characteristics to a collaborative-based method (or vice versa) [12], or by combining predictions obtained separately using a content-based method and a CF method [88], or by model unification [97, 14]. Other hybrid methods include Fab which makes use of profiles information to determine similar users for CF [12], combination of CF and content-based approaches using the prediction strengths [88], probabilistic mixture models [97], a kernel-based method which allows generalization across the user and item dimensions simultaneously [14]. [19] surveys the area of possible hybrid recommender systems and examines different types of combinations.

1.2.4 Evaluation Metrics

Recommender systems have been evaluated in many, often incomparable, ways [51]. The work of [61, 51] review several different metrics of predictive accuracy, coverage, learning rate, novelty and serendipity, and confidence. In this part, we review some key metrics which will be used in evaluating the rating prediction and *Top-N* recommendation of collaborative filtering in Chapter 3.

For evaluation of rating prediction, prediction accuracy is by far the most discussed characteristic of a recommendation system. There are much work which focuses on evaluating the accuracy property of a system [51]. Prediction accuracy empirically measures how close the predicted ratings of a system differs from the given ratings in the average sense, or for each user how well a system's predicted ranking of items suits the given ranking order of items. Predictive accuracy metrics and ranking accuracy metrics are two significant classes for evaluation of prediction accuracy.

For evaluating predictive accuracy, Mean Absolute Error (**MAE**) and Root Mean Square Error (**RMSE**) [65, 66, 102, 51] are two classic and widely-adopted metrics.

The formal definitions of these two metrics are as follows

$$MAE = \frac{\sum_{(u,i)\in R_t} |r_{ui} - \hat{r}_{ui}|}{|R_t|}$$

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in R_t} (r_{ui} - \hat{r}_{ui})^2}{|R_t|}}$$

where values of these two metrics close to zero show better performance, and $RMSE$ tends to penalize larger errors more severely than MAE.

As mentioned in [51], "ranking accuracy metrics can be used to evaluate the ability of a recommendation algorithm to produce a recommended order of items that matches how the user would have ordered the same items". The average Discounted Cumulative Gain (DCG) and Normalized Discounted Cumulative Gain ($NDCG$) [79, 130] of all test users are the most commonly-adopted measures for the ranking accuracy. The formal definitions of average DCG and $NDCG$ of all users are

$$DCG@N = \frac{1}{|\mathcal{U}|} \sum_{u=1}^{|\mathcal{U}|} \sum_{n=1}^{N} \frac{2^{r_u^n} - 1}{\log_2(n+1)}$$

$$NDCG@N = \frac{1}{|\mathcal{U}|} \sum_{u=1}^{|\mathcal{U}|} \frac{1}{Z_u} \sum_{n=1}^{N} \frac{2^{r_u^n} - 1}{\log_2(n+1)}$$

where r_u^n is the ground truth rating value of the item at the position n predicted by the algorithms. $\frac{1}{\log_2(n+1)}$ is a position discount factor. Highly relevant items appearing lower in the recommendation list will be penalized by the discount factor. $|\mathcal{U}|$ indicates the number of users in the test dataset, Z_u is the maximum value of $\sum_{n=1}^{N} \frac{2^{r_u^n}-1}{\log_2(n+1)}$ for the user u and works as a normalization factor of this user. Since the $NDCG$ is normalized, it takes a value from 0 to 1.

Top-N recommendation is another fundamental property of recommenders [61, 58]. Evaluation of *Top-N* recommendation aims to evaluate whether a set of items are the most appealing to a particular user. For evaluation of *Top-N* recommendation, the overall *Recall* [65, 82, 102] is the most widely-used measure, which is computed by averaging over all users. The *Recall* metric is expressed as follows

$$Recall@N = \frac{1}{|\mathcal{U}|}\sum_{u=1}^{|\mathcal{U}|}\frac{\#hits(u,N)}{|\mathcal{U}|}$$

where $\#hits(u,N)$ means whether the u's most appealing item in the test dataset appears in the predicted list within the length N. *Recall* value increases with the length N and the maximum value is 1. Larger *Recall* values indicate better *Top-N* recommendation.

1.3 Contribution of this Thesis

This thesis studies two fundamental problems of data mining, which are outlier detection from large-scale categorical datasets and recommendation systems from highly-skewed rating datasets. The contributions of this thesis are twofold and will be briefly summarized in the next two subsections.

1.3.1 Contributions to Outlier Detection

For outlier detection from categorical data, over the years, a number of methods have been developed [75, 21, 37, 49, 92]. However, real-world datasets and environments present a range of difficulties that limit the effectiveness of these methods. Existing methods for outlier detection from categorical datasets suffer from the following limitations:

1. First of all, there does not exist a formal definition of categorical outlier in the literature. Many methods are based on definitions of the numerical outliers, which usually cannot well reflect the characteristic of categorical outliers. Without a formal definition, methods of categorical outlier detection are often designed as an ad-hoc process.

2. Many existing methods suffer from low effectiveness and low efficiency due to

high dimensionality and large size of the dataset, high-complexity of statistical tests or inefficient proximity-based measures. For instance, the distance-based categorical outlier detection methods, such as CNB [75], are very time-consuming for large datasets. The time complexity of CNB increases quadratically with the number of objects. The time costs of rule-based methods, FIB [49] and OA [92], also increase quadratically with the number of attributes.

3. Many methods for detecting categorical outliers [75, 49, 92], requires that the user provides parameters to measure whether an object possesses properties sufficiently different from others to be qualified as an outlier. The performances of these methods are heavily dependent on parameter settings, which are very difficult to estimate without background knowledge about the data.

In this thesis, we propose information-theory-based effective outlier detection methods for large categorical datasets. Our work addresses the existing limitations mentioned above. First of all, we deal with the lack of a formal definition of outlier by using information theory. The proposed definition helps to construct general outlier detection methods for categorical datasets. In fact, based on this definition, we propose an optimization-based model for detecting outliers of categorical datasets, where a novel concept of weighted holo-entropy is utilized to capture the distribution and correlation information of a dataset. To avoid high time-complexity, we derive a new outlier factor function from the objective function and show that computation/updating of the outlier factor is solely determined by the object itself and can be performed efficiently without the need to estimate the joint probability distribution. Our proposed methods have a linear time complexity with the size of datasets, i.e. number of objects and dimensions of the datasets, and need only the number of outliers as an input parameter.

1.3.2 Contributions to Recommendation Systems

Primarily, existing methods for recommendation systems take into account the rating datasets of movie or music, e.g. Netflix, Movielens, EachMovie and Yahoo Music datasets. In the rating scenario of these rating datasets, users are prone to choose movies or musics belonging to some genres matching their interests, and provide most objective ratings to these items while expecting the system to become more adapted to recommend most appealing items to them. The ratings of these datasets are more concentrated around the middle of the rating range, and are more likely symmetric and evenly distributed on both side of the mean rating.

However, there are also many rating datasets with skewed distributions which are very different from the distribution of the above mentioned datasets. These skewed rating datasets broadly exist in e-commerce and content provider websites, e.g. Amazon, Epinions and Youtube [24]. The users on the e-commerce websites tend to give the highest ratings to the desired products, after they may have compared these products to other similar ones. On the other hand, on the content provider websites, the users are prone to provide most positive feedback towards a small portion of most-appealing items in order to express their opinion or influence others' choice. In these cases, the ratings are likely with higher asymmetry and majority of ratings are in the highest side of rating range.

In the existing methods of CF, the non-transitive correlations among users or items are also a problem. If two similar items have never been rated by the same user, or two similar users have never rated the same item, the similarity relationship may not be well captured by existing methods. In these cases, the non-transitivity associations may affects the performance of the system.

To deal with these problems, we propose a new framework for estimating the rating and quantitative high-order preference simultaneously. This framework allows to create novel and efficient models for skewed rating datasets. It relies on high-order quantitative preference of users to better capture the users' relative rating information among items. At the same time, the transitive associations among the items which are never rated together can be implicitly captured by the constraints of high-order preference similarity. New models created under this framework can generate better performance than the conventional methods on the skewed rating datasets for not only rating prediction but also for *Top-N* recommendation.

As evidence of the contributions, here is the list of the author's published or submitted papers issued from his work in relation with this thesis.

- S. Wu and S. Wang, "A Quantitative High-order Preference Framework for High-skewed Rating Datasets", submitted, 2012.

- H. Ni, B. Abdulrazak, D. Zhang, S. Wu, X. Zhou, K. Miao and D. Han, "Multi-modal Non-intrusive Sleep Pattern Recognition in Elder Assistive Environment", In *Proceedings of the 10th International Conference on Smart Homes and Health Telematics (ICOST 2012).*

- S. Wu and S. Wang, "Information-theoretic Outlier Detection for large-scale Categorical Data", *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, Preprint.
`http://www.computer.org/portal/web/csdl/doi/10.1109/TKDE.2011.261`

- H. Ni, B. Abdulrazak, D. Zhang, S. Wu, Z. Yu, X. Zhou, S. Wang, "Towards Non-intrusive Sleep Pattern Recognition in Elder Assistive Environment", *Journal of Ambient Intelligence and Humanized Computing (AIHC)*, Springer, 2012.
`http://www.springerlink.com/content/x4q1824375263515/`

- S. Wu and S. Wang, "Parameter-free Anomaly Detection for Categorical Data", In *Proceedings of the 7th International Conference on Machine Learning and Data Mining (MLDM 2011)*, 2011.

- H. Ni, B. Abdulrazak, D. Zhang, S. Wu, "CDTOM: A Context-driven Task-oriented Middleware for Pervasive Homecare Environment", *International Journal of UbiComp (IJU)*, Vol.2, No.1, Jan. 2011.
`http://arxiv.org/ftp/arxiv/papers/1102/1102.1152.pdf`

- S. Wu and S. Wang, "Rating-based Collaborative Filtering Combined with Additional Regularization", In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2011)*, 2011.

- H. Ni, B. Abdulrazak, D. Zhang, S. Wu, Z. Yu, X. Zhou, S. Wang, "Towards Non-intrusive Sleep Pattern Recognition in Elder Assistive Environment", In

Proceedings of the 7th International Conference Ubiquitous Intelligence and Computing (UIC 2010), 2010.

- H. Ni, B. Abdulrazak, D. Zhang and S. Wu, "Unobtrusive Sleep Posture Detection for Elder-Care in Smart Home", In *Proceedings of the 8th International Conference on Smart Homes and Health Telematics (ICOST 2010)*, 2010.

Chapter 2

Outlier Detection in Large-scale Categorical Data

In this Chapter, we are investigating outlier detection for categorical datasets. We have formulated outlier detection as an optimization problem and proposed two practical, unsupervised, 1-parameter algorithms for detecting outliers in large-scale categorical datasets. The effectiveness of our algorithms results from a new concept of weighted holo-entropy that considers both the data distribution and attribute correlation to measure the likelihood of outlier candidates. The efficiency of our algorithms results from the outlier factor function derived from the holo-entropy. A new outlier factor function is derived from the optimization function and show that computation/updating of the outlier factor is solely determined by the object itself and can be performed efficiently without the need to estimate the joint probability distribution. Besides, We also estimate an upper bound for the number of outliers and an anomaly candidate set. This bound, obtained under a very reasonable hypothesis on the number of possible outliers, allows us to further reduce the search cost. The proposed algorithms have been evaluated on real and synthetic datasets. Our experiments in comparison with other algorithms confirm the effectiveness and efficiency of the proposed algorithms in practice.

The included paper has been accepted by *IEEE Transactions on Knowledge and Data Engineering*, 16th Dec. 2011.

Information-theoretic Outlier Detection for Large-scale Categorical Data

Shu Wu and Shengrui Wang[1]

Abstract

Outlier detection can usually be considered as a pre-processing step for locating, in a dataset, those objects that do not conform to well-defined notions of expected behavior. It is very important in data mining for discovering novel or rare events, anomalies, vicious actions, exceptional phenomena, etc. We are investigating outlier detection for categorical datasets. This problem is especially challenging because of the difficulty of defining a meaningful similarity measure for categorical data. In this paper, we propose a formal definition of outliers and an optimization model of outlier detection, via a new concept of holo-entropy that takes both entropy and total correlation into consideration. Based on this model, we define a function for the outlier factor of an object which is solely determined by the object itself and can be updated efficiently. We propose two practical 1-parameter outlier detection methods, named ITB-SS and ITB-SP, which are capable to identify the most likely outliers automatically. Users need only to provide the number of outliers they want to detect. Experimental results show that ITB-SS and ITB-SP are more effective and efficient than mainstream methods and can be used to deal with both large and high-dimensional datasets where existing algorithms fail.

Keywords: Outlier detection, Holo-entropy, Total correlation, Outlier factor, Attribute weighting, Greedy algorithms

[1] The authors are with the Department of Computer Science, University of Sherbrooke, Sherbrooke, QC J1K 2R1, Canada. E-mail: {shu.wu, shengrui.wang}@usherbrooke.ca.

2.1 Introduction

Outlier detection, which is an active research area [22, 52, 23, 117], refers to the problem of finding objects in a dataset that do not conform to well-defined notions of expected behavior. The objects detected are called outliers, also referred to as anomalies, surprises, aberrants, etc. Outlier detection can be implemented as a preprocessing step prior to the application of an advanced data analysis method. It can also be used as an effective tool to discover interest patterns such as the expense behavior of a to-be-bankrupt credit cardholder. Outlier detection is an essential step in a variety of practical applications including intrusion detection [71], health system monitoring [52] and criminal activity detection in E-commerce [8], and can also be used in scientific research for data analysis and knowledge discovery in biology, chemistry, astronomy, oceanography and other fields. [52].

According to [22][52], if the existing methods for outlier detection are classified according to the availability of labels in the training datasets, there are three broad categories: supervised, semi-supervised and unsupervised approaches. In principle, models within the supervised or the semi-supervised approaches all need to be trained before use, while models adopting the unsupervised approach do not include the training phase. Moreover, in a supervised approach a training set should be provided with labels for anomalies as well as labels of normal objects, in contrast with the training set with normal object labels alone required by the semi-supervised approach. On the other hand, the unsupervised approach does not require any object label information. Thus the three approaches have different prerequisites and limitations, and they fit different kinds of datasets with different amounts of label information. The three broad categories of outlier detection techniques are discussed below.

The *supervised anomaly detection approach* learns a classifier using labeled objects belonging to the normal and anomaly classes, and assigns appropriate labels to test objects. The supervised approach has been studied extensively and many methods have been developed. For instance, the group of proximity-based methods includes the cluster-based 'K-Means+ID3' algorithm [39], which cascades K-Means clustering and an ID3 decision tree for classifying anomalous and normal objects. The work of [13] is based on statistical testing and an application of Transduction Confidence Machines, which requires k neighbors. Moreover, one-class SVMs [119][108] have been applied broadly in this field as they do not have to make a probability density estimation. A variety of methods [35][57] based on information theory have also been proposed. The work of [35] proposes a method to control the false positive rate in the novelty detection problem. In [57], a formal Bayesian definition of surprise is proposed.

The *semi-supervised anomaly detection approach* primarily learns a model representing normal behavior from a given training dataset of normal objects, and then calculates the likelihood of a test object's being generated by the learned model. Zhang [134] proposes an adapted hidden Markov model for this approach to anomaly detection, while Gao [40] proposes a clustering-based algorithm which punishes deviation from known labels. Methods that assume availability of only the outlier objects for training are rare [52], because it is difficult to obtain a training dataset which covers all possible abnormal behavior that can occur in the data.

The *unsupervised anomaly detection approach* detects anomalies in an unlabeled dataset under the assumption that the majority of the objects in the dataset are normal. Angiulli et al. [10] propose a KNN distance-based method. Clustering is another widely implemented method, of which [6] is an example. Moreover, this approach is applied to different kinds of outlier detection tasks and datasets, e.g., conditional anomaly detection [110], context-aware outliers [127] and outliers in semantic graphs [76]. As this approach does not require a labeled training dataset and is suitable for different outlier detection tasks, it is the most widely used.

To implement supervised and semi-supervised outlier detection methods, one must first label the training data. However, when faced with a large dataset with millions of high-dimensional objects and a low anomalous data rate, picking the abnormal and normal objects to compose a good training dataset is time-consuming and labor-intensive. The unsupervised approach is important not only for its low requirement in terms of a priori knowledge about the outliers but also for the role of preprocessing it can play. For instance, in a supervised approach, an unsupervised method can be used as the first step to find a candidate set of outliers, which will help experts to build the training dataset. The unsupervised approach is our research focus in this paper.

2.1.1 Unsupervised Categorical Outlier Detection

In real applications, a large portion or the entirety of the dataset is often presented in terms of categorical attributes. Examples of such datasets include transaction data, financial records in commercial banks, demographic data, etc. The problem of outlier detection in this type of dataset is more challenging since there is no inherent measurement of distance between the objects. Existing unsupervised outlier detection methods, e.g. LOF [17], LOCI [93] and [6][11], are effective on datasets with numerical attributes. However they cannot be easily adapted to deal with categorical data.

Outlier detection methods for categorical data can be characterized by the way outlier candidates are measured w.r.t. other objects in the dataset. In general, outlier candidates can be assessed based either on data distribution or on attribute correlation, which provides a more global measure. They can also be assessed using a between-object similarity or local density, which provides a local measure. Various techniques such as proximity-based [75], rule-based [49], and information-theoretic [72] methods have been proposed (Section 2 provides a more detailed discussion) and fall into one of these two categories. The common problem with the existing methods is the lack of a formal definition for the outlier detection problem. Without a formal definition, outlier detection is often designed as an ad-hoc process. In particular, several user-defined parameters are often required to define whether an object possesses properties sufficiently different from others to be qualified as an outlier. Such methods are heavily dependent on parameter settings, which are very difficult to estimate without background knowledge about the data. Many existing methods also suffer from low effectiveness and low efficiency due to high dimensionality and large size of the dataset, high-complexity statistical tests or inefficient proximity-based measures.

2.1.2 Objectives and Contributions

The goal of this paper is twofold. First, we deal with the lack of a formal definition of outliers and modeling of the outlier detection problem; second, we aim to propose effective and efficient methods that can be used to solve the outlier detection problem in real applications. In this paper, these two goals are achieved by exploring the information-theoretic approach [25].

First, in our approach, we adopt the deviation-based strategy which, according to [43], avoids the use of statistical tests and proximity-based measures to identify exceptional objects. We explore information theory [25] to derive several new concepts. In particular, we combine entropy and total correlation with attribute weighting to define the concept of weighted holo-entropy, where the entropy measures the global disorder of a data set and the total correlation measures the attribute relationship. Based on this concept, we build a formal model of outlier detection and propose a criterion for estimating the "goodness" of a subset of objects as potential outlier candidates. Then outlier detection is formulated as an optimization problem involving searching for the optimal subset in terms of "goodness" and number of outliers. Finally, to solve the optimization problem, we carry out a deep investigation of the analytical and statistical properties of the proposed criterion and propose two greedy algorithms that effectively bypass probability estimation and the high complexity of exploring the whole outlier candidate space.

The contributions of this work are as follows.

1. We propose a formal optimization-based model of categorical outlier detection, for which a new concept of weighted holo-entropy which captures the distribution and correlation information of a dataset is proposed.

2. To solve the optimization problem, we derive a new outlier factor function from the weighted holo-entropy and show that computation/updating of the outlier factor can be performed without the need to estimate the joint probability distribution. We also estimate an upper bound of outliers to reduce the search space.

3. We propose two effective and efficient algorithms, named the Information-Theory-Based Step-by-Step (**ITB-SS**) and Single-Pass (**ITB-SP**) methods. These algorithms need only the number of outliers as an input parameter and completely dispense with the parameters for characterizing outliers usually required by existing algorithms.

The rest of this paper is organized as follows. Section 2 discusses related work and gives a detailed description of the methods which will be compared. Section 3 presents the concepts of holo-entropy and modeling of outlier detection as an optimization problem. Section 4 describes the proposed algorithms for solving the detection problem. Major experimental results, including comparisons with existing methods, are presented in Section 5. Section 6 discusses a potentially interesting avenue for developing a true parameter-free detection algorithm. The conclusion is given in Section 7.

2.2 Related Work

Mainstream methods/algorithms designed for outlier detection from categorical data can be grouped into four categories. Some of these algorithms are compared with the proposed algorithms in Section 2.5.

2.2.1 Proximity-based Methods

Being intuitively easy to understand, proximity-based outlier detection, which measures the nearness of objects in terms of distance, density, etc, is an important technique adopted by many outlier detection methods. For numerical outlier detection, there are a variety of methods [64][17][10][15] in this category. For instance, LOF [17] is an effective method that utilizes a concept of local density to measure how isolated an object is w.r.t. the surrounding *Minpts* objects.

For categorical datasets, the proximity-based methods must confront the problems of how to choose the measurement of distance or density and how to avoid high time and space complexity in the distance computing process. For instance, ORCA [15] uses the Hamming distance and CNB [75] employs a common-neighbor-based distance to measure the distance between categorical objects. Let us have a closer look at the CNB algorithm. It consists of two steps, the neighbor-set generating step and the outlier mining step. The neighbor-set of the k nearest neighbors with similarity threshold θ to all objects is computed in the neighbor-set generation step. Both k and θ are user-defined parameters. In the second step, an outlier factor for each object is computed by summing its distance from its neighbors. The objects with the o (number of outliers) largest values are set to be outliers. The proximity-based approach has many prerequisite parameters, which need repeated trial-and-error to attain the desired result. Proximity-based methods also suffer from the curse of dimensionality when using distance or local density measures on the full dimensions. In general, these methods are time- and space-consuming and consequently are not appropriate for large datasets.

2.2.2 Rule-based Methods

Rule-based methods borrow the concept of frequent items from association-rule mining. Such methods consider the frequent or infrequent items in the dataset. For instance, in the work of [21][37], objects with few frequent items or many infrequent items are more likely to be considered as anomalous objects than others.

Frequent Pattern Outlier Factor (called the *FIB* method in this paper) [49] and Otey's Algorithm (called the *OA* method in this paper) [92] are two well-known rule-based techniques. The procedure of the FIB algorithm includes an initial computation of the set of frequent patterns, using a pre-defined minimum support threshold. For each object, all support values of associated frequent patterns are summed up as the outlier factor of this object. The objects with the o smallest factors are considered as the outliers. Contrary to the FIB algorithm, OA begins by collecting the infrequent items from the dataset. Based on the infrequent items, the outlier factors of the objects are computed. The objects with the o largest scores are treated as outliers. The time complexity of both algorithms is determined by the frequent-item or infrequent-item generating processes. For instance, the time complexity of the FIB method is exponentially increasing with the number of attributes due to the *Apriori* algorithm [7]. Therefore, this approach is limited to low-dimensional datasets.

2.2.3 Information-theoretic Methods

Several information-theoretic methods have been proposed in the literature. For anomaly detection in audit datasets, Lee [72] presents a series of information-theoretic measures, i.e., entropy, conditional entropy, relative conditional entropy and information gain, to identify outliers in the univariate audit dataset, where the attribute relationship does not need to be considered. The work of [48] employs entropy to measure the disorder of a dataset with the outliers removed. In these methods, heuristic local search is used to minimize the objective function. The methods proposed in [28][29] set a threshold of mutual information and obtain a set of dependent attribute pairs. Based on this set, an outlier factor for each individual object is defined. In general, information-theoretic methods focus either on a single entropy-like measurement or on mutual information, and require expensive estimation of the joint probability distribution when the dataset is shrunk following elimination of certain outliers.

2.2.4 Other Methods

Several other approaches using the Random Walk, Hypergraph theory or clustering methods have been proposed to deal with the problem of outlier detection in categorical data. For instance, based on hypergraph theory, HOT [129] captures the distribution characteristics of an object in the subspaces and these characteristics are then used to identify outliers. In the random-walk-based method [89], outliers are those objects with a low probability of jumping to neighbors. In other words, they have a high probability of staying in their states. In [132], the relationships among the neighbors are considered and a mutual-reinforcement-based local outlier factor is proposed to identify outliers. This can also be viewed as a random-walk method with a fixed number of walk steps. In [47], a cluster-based local outlier detection method is proposed to identify the physical significance of an object. The outlier factor in this method is measured by both the size of the cluster the object belongs to and the distance between the object and its closest cluster. These methods are not very efficient for large or high-dimensional datasets because they contain some high-complexity procedures, e.g., frequent itemsets generating processes in HOT [129], similarity computation in the random-walk-based methods [89][132], and the clustering process in the cluster-based method [47].

2.3 Measurement for Outlier Detection

In this section, we first look at how entropy and total correlation can be used to capture the likelihood of outlier candidates. We propose the concept of holo-entropy and formulate the outlier detection problem.

2.3.1 Entropy and Total Correlation

Consider a set $\mathcal{X}$ containing n objects $\{x_1, x_2, ..., x_n\}$, where each x_i for $1 \leq i \leq n$ is a vector of categorical attributes $[y_1, y_2, ..., y_m]^T$, where m is the number of attributes, y_j has a value domain determined by $[y_{1,j}, y_{2,j}, ..., y_{n_j,j}]$ $(1 \leq j \leq m)$ and n_j indicates the number of distinct values in attribute y_j. Considering each y_j as a random variable, the random vector $[y_1, y_2, ..., y_m]^T$ is represented by $\mathcal{Y}$. x_i can be denoted as $(x_{i,1}, x_{i,2}, ..., x_{i,m})^T$. We use $H_{\mathcal{X}}()$, $I_{\mathcal{X}}()$ and $C_{\mathcal{X}}()$, respectively, to represent entropy, mutual information and total correlation computed on the set $\mathcal{X}$; e.g., $I_{\mathcal{X}}(y_i; y_j)$ represents the mutual information between attributes y_i and y_j. Sometimes, we drop off the index term $\mathcal{X}$ when there is no ambiguity, e.g., using $I(y_i; y_j)$ in place of $I_{\mathcal{X}}(y_i; y_j)$.

Now, based on the chain rule for entropy [25], the entropy of $\mathcal{Y}$, denoted as $H_{\mathcal{X}}(\mathcal{Y})$ can be written as follows:

$$\begin{aligned} H_{\mathcal{X}}(\mathcal{Y}) = H_{\mathcal{X}}(y_1, y_2, ..., y_m) &= \sum_{i=1}^{m} H_{\mathcal{X}}(y_i | y_{i-1}, ..., y_1) \\ &= H_{\mathcal{X}}(y_1) + H_{\mathcal{X}}(y_2|y_1) + ... + H_{\mathcal{X}}(y_m|y_{m-1}, ..., y_1) \end{aligned} \tag{2.1}$$

where $H_{\mathcal{X}}(y_m|y_{m-1},...,y_1) = -\sum_{y_m,y_{m-1},...,y_1} p(y_m,y_{m-1},...,y_1)\text{log}p(y_m|y_{m-1},...,y_1)$.

The entropy can be used as a global measure in outlier detection. In information theory, entropy means uncertainty relative to a random variable: if the value of an attribute is unknown, the entropy of this attribute indicates how much information we need to predict the correct value. A subset of objects are good outlier candidates if their removal from the data set causes significant decrease of the entropy of the data set. The method proposed in [72] makes use of entropy as a quality measure in outlier detection from unidimensional audio data. He et al. [48] extend this schema to measure the disorder of a multi-dimensional dataset with the outliers removed, where a heuristic local search is employed to minimize the objective function.

Let us look at how total correlation can also be used in outlier detection. The total correlation [113] is defined as the sum of mutual information of multivariate discrete random vectors $\mathcal{Y}$, denoted as $C_{\mathcal{X}}(\mathcal{Y})$.

$$\begin{aligned} C_{\mathcal{X}}(\mathcal{Y}) &= \sum\nolimits_{i=2}^{m} \sum\nolimits_{\{r_1...r_i\}\subset\{1,...,m\}} I_{\mathcal{X}}(y_{r_1};...;y_{r_i}) \\ &= \sum_{\{r_1,r_2\}\subset\{1,...,m\}} I_{\mathcal{X}}(y_{r_1};y_{r_2}) + ... + I_{\mathcal{X}}(y_{r_1};...;y_{r_m}) \end{aligned} \tag{2.2}$$

where $r_1...r_i$ are attribute numbers chosen from 1 to m. $I_{\mathcal{X}}(y_{r_1};...;y_{r_i}) = I_{\mathcal{X}}(y_{r_1};...;y_{r_{i-1}})$ $- I_{\mathcal{X}}(y_{r_1};...;y_{r_{i-1}}|y_{r_i})$ [25] is the multivariate mutual information of $y_{r_1}...y_{r_i}$, where $I_{\mathcal{X}}(y_{r_1};...;y_{r_{i-1}}|y_{r_i}){=}E(I(y_{r_1};...;y_{r_{i-1}})|y_{r_i})$ is the conditional mutual information. The total correlation is a quantity that measures the mutual dependence or shared information of a dataset.

Taking the case of total correlation $C_{\mathcal{X}}(y_1;y_2)$ with two attributes y_1 and y_2 as an example, $C_{\mathcal{X}}(y_1;y_2) = I_{\mathcal{X}}(y_1;y_2)$ denotes the total correlation for a random vector $\mathcal{Y}$ with two attributes y_1 and y_2. Its value corresponds to the reduction in the uncertainty of one attribute value yielded by knowledge of the other. If the value of $C_{\mathcal{X}}(y_1;y_2)$ is large, it means that the number of duplicate pairs of attribute values is small in these two attributes compared with the situation when the value of $C_{\mathcal{X}}(y_1;y_2)$ is small. In general, for the case where there are more than two attributes, larger $C_{\mathcal{X}}(\mathcal{Y})$

means a smaller number of objects sharing common attribute values, which in turn implies fewer number of frequent itemsets and worse cluster structure. Thus, similar to entropy, the total correlation can be used to measure the goodness of the outlier candidates in a subset $\mathcal{O}$ by evaluating $C_{\mathcal{X}\backslash\mathcal{O}}(\mathcal{Y})$. Again, the smaller the value of $C_{\mathcal{X}\backslash\mathcal{O}}(\mathcal{Y})$, the better the subset $\mathcal{O}$ as a set of outlier candidates.

2.3.2 Holo-entropy

We begin here with an example to show that entropy alone is not a good enough measure for outlier detection and the contribution of the total correlation is necessary. Looking at the example in Table 2.1, where 14 objects with four attributes are illustrated, we represent the dataset by $\mathcal{X}$. $\mathcal{X}$ includes two objects x_{13} and x_{14} which can be identified as the most likely outliers by comparison with the other 12 objects. Moreover, x_{14} is clearly more exceptional than x_{13} since it shares none of its attributes with the rest of objects. Now, $H_{\mathcal{X}\backslash x_{14}}(\mathcal{Y})$=$H_{\mathcal{X}\backslash x_{13}}(\mathcal{Y})$=3.7 means that, if only the entropy is used, x_{14} and x_{13} are equally exceptional as outlier candidates. On the other hand, if we combine the total correlation and the entropy, we obtain $H_{\mathcal{X}\backslash x_{14}}(\mathcal{Y})$+$C_{\mathcal{X}\backslash x_{14}}(\mathcal{Y})$=9.414 and $H_{\mathcal{X}\backslash x_{13}}(\mathcal{Y})$+$C_{\mathcal{X}\backslash x_{13}}(\mathcal{Y})$=10.030, which allows object x_{14} to be distinguished as a more likely outlier than x_{13}. Interestingly, given the distributions of attributes in a dataset, there is a complementary relationship that exists between the entropy and total correlation of $\mathcal{Y}$. It is based on Watanabe's proof [128] that the total correlation can be expressed as $C_{\mathcal{X}}(\mathcal{Y}) = \sum_{i=1}^{m} H_{\mathcal{X}}(y_i) - H_{\mathcal{X}}(\mathcal{Y})$. This motivates the following definition of holo-entropy as a new measure for outlier detection.

Definition 7. (Holo-entropy of a Random Vector) *The holo-entropy $HL_{\mathcal{X}}(\mathcal{Y})$ is defined as the sum of the entropy and the total correlation of the random vector $\mathcal{Y}$, and can be expressed by the sum of the entropies on all attributes.*

$$HL_{\mathcal{X}}(\mathcal{Y}) = H_{\mathcal{X}}(\mathcal{Y}) + C_{\mathcal{X}}(\mathcal{Y}) = \sum\nolimits_{i=1}^{m} H_{\mathcal{X}}(y_i) \qquad (2.3)$$

Note that when the components of $\mathcal{Y}$ are independent or $\mathcal{Y}$ has only one component, $HL_{\mathcal{X}}(\mathcal{Y}) = H_{\mathcal{X}}(\mathcal{Y})$, i.e., the holo-entropy coincides with the entropy.

Table 2.1: Adjusting Total Correlation

# Object	y_1	y_2	y_3	y_4
x_1	a_1	a_2	a_3	a_4
x_2	a_1	a_2	b_3	b_4
x_3	a_1	a_2	c_3	c_4
x_4	a_1	a_2	d_3	d_4
x_5	a_1	a_2	e_3	e_4
x_6	a_1	a_2	f_3	f_4
x_7	b_1	b_2	g_3	g_4
x_8	c_1	c_2	g_3	g_4
x_9	d_1	d_2	g_3	g_4
x_{10}	e_1	e_2	g_3	g_4
x_{11}	f_1	f_2	g_3	g_4
x_{12}	g_1	g_2	g_3	g_4
x_{13}	b_1	d_2	c_3	a_4
x_{14}	h_1	h_2	h_3	h_4

The example in Fig. 2.1 illustrates how holo-entropy is more appropriate than entropy or total correlation for describing outliers. Fig. 2.1(a) is the original dataset containing 6 objects, in which the object (b_1, c_2) and to a lesser extent the object (a_1, b_2) are most likely to be outliers. Fig. 2.1(b), 2.1(c) and 2.1(d) illustrate three possible datasets which result when one object is removed. Similar to the example in Table 2.1, Fig. 2.1(c) and Fig. 2.1(d) show that entropy provides no hint as to which one, (b_1, c_2) or (a_1, b_2), is more likely to be an outlier. On the other hand, if only the total correlation is taken into consideration, Fig. 2.1(c) indicates the smallest total correlation for $C_{\mathcal{X}\setminus\{(b_1\ c_2)\}}(\mathcal{Y})$ for (b_1, c_2), while Fig. 2.1(b) and Fig. 2.1(d) indicate that (a_1, a_2) and (a_1, b_2) are equally likely to be outliers, which is wrong. The holo-entropy allows us to clearly establish appropriate outlier likelihoods among (b_1, c_2), (a_1, b_2) and (a_1, a_2).

Proposition 1. $0 \leq HL_{\mathcal{X}}(\mathcal{Y}) \leq m \log(n)$

Proof. For an attribute y_i of $\mathcal{Y}$, if all its values are the same, the minimum entropy of this attribute satisfies $H_{\mathcal{X}}(y_i)$=0. If all the values of y_i are different, the maximum

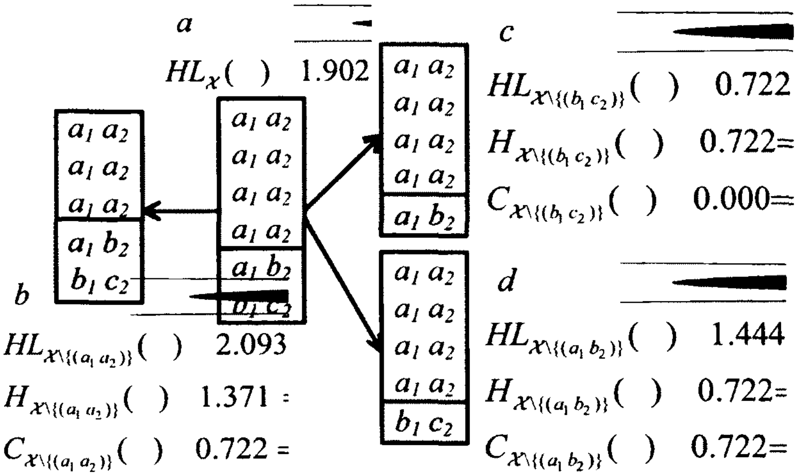

a
$HL_{\mathcal{X}}(\ )$ 1.902
b
$HL_{\mathcal{X}\setminus\{(a_1,a_2)\}}(\ )$ 2.093
$H_{\mathcal{X}\setminus\{(a_1,a_2)\}}(\ )$ 1.371
$C_{\mathcal{X}\setminus\{(a_1,a_2)\}}(\ )$ 0.722
c
$HL_{\mathcal{X}\setminus\{(b_1,c_2)\}}(\ )$ 0.722
$H_{\mathcal{X}\setminus\{(b_1,c_2)\}}(\ )$ 0.722
$C_{\mathcal{X}\setminus\{(b_1,c_2)\}}(\ )$ 0.000
d
$HL_{\mathcal{X}\setminus\{(a_1,b_2)\}}(\ )$ 1.444
$H_{\mathcal{X}\setminus\{(a_1,b_2)\}}(\ )$ 0.722
$C_{\mathcal{X}\setminus\{(a_1,b_2)\}}(\ )$ 0.722
$a_1\ a_2$
$a_1\ b_2$
$b_1\ c_2$

Figure 2.1: Entropy, Total Correlation and Holo-entropy for Outlier Detection

entropy is noted as $H_{\mathcal{X}}(y_i) = \log(n)$. Since $HL_{\mathcal{X}}(\mathcal{Y}) = \sum_{i=1}^{m} H_{\mathcal{X}}(y_i)$, the inequalities hold. $\square$

2.3.3 Attribute Weighting

The proposed holo-entropy assigns equal importance to all the attributes, whereas in real applications, different attributes often contribute differently to form the overall structure of the dataset. In this section, after demonstrating the need for attribute weighting, we will propose a simple method for weighting attributes and then modify the holo-entropy by incorporating the attribute weights. The proposed weighting method computes the weights directly from the data and is motivated by increased effectiveness in practical applications rather than by theoretical necessity. In the outlier detection algorithms proposed in Section 2.4, the attributes are assumed to be weighted. The "unweighted" version of the proposed algorithms can be obtained simply by setting all the weights to one. In Section 2.5, both weighted and unweighted algorithms are evaluated.

As an example, let us look at the data from a survey on positive attitude towards science given in Table 2.2, where the observations (surveyed persons) are described by their education level and age range. We will argue that for outlier detection from this survey data, the attribute *Degree* is more important than the attribute *Age*.

Table 2.2: Weighted Holo-entropy in Outlier Detection

# Case	*Degree*	*Age*	$HL_{\mathcal{X}\setminus\{x_o\}}(\mathcal{Y})$	$\mathcal{W}_{\mathcal{X}\setminus\{x_o\}}(\mathcal{Y})$
1	*Master's*	$[30,40)$	3.507	1.050
2	*Master's*	$[30,40)$	3.507	1.050
3	*Master's*	$[30,40)$	3.507	1.050
4	*High School*	$[30,40)$	**3.113**	**0.895**
5	*Ph.D.*	$[20,30)$	**3.113**	0.967
6	*Ph.D.*	$[40,50)$	**3.113**	0.967
7	*Ph.D.*	$[50,60)$	**3.113**	0.967
8	*Ph.D.*	$[60,70)$	**3.113**	0.967

According to the column $HL_{\mathcal{X}\setminus\{x_o\}}(\mathcal{Y})$ in Table 2.2, the cases 4,5,6,7 and 8 are equally likely to be outliers since the removal of each results in the same decrease in the value of $HL_{\mathcal{X}\setminus\{x_o\}}(\mathcal{Y})$. In fact, each of the cases 4,5,6,7 and 8 is distinguished by its value on either the *Degree* or the *Age* attribute. By looking at the internal structure of the values of each attribute, we see that *High-School* is more outstanding within *Degree* than, for example, [40, 50) is within *Age*, since [40, 50) is one of the four values that are different from the dominating value [30,40), while *High-School* is the only value different from the dominating values *Master* and *Ph.D.* In other words, it is the good cluster structure of the attribute *Degree*, compared to that of *Age*, that makes *High-School* more outstanding than [40,50). The weighting strategy proposed in this paper aims to give more importance to the attribute *Degree* so that the case (*High-School*, [30,40)) is identified as a more likely outlier candidate than, for example, the case (*Ph.D.*, [40,50)).

Given that the holo-entropy is defined as the sum of entropies of individual attributes and outliers are detected by minimizing the holo-entropy through the removal of outlier candidates, our strategy consists in weighting the entropy of each individual attribute in order to give more importance to those attributes with small entropy values, e.g. *Degree* in the example of Table 2.2. This increases the impact of removing an outlier candidate that is outstanding on those attributes. To weight the entropy of each attribute, we propose to employ a reverse sigmoid function of the entropy, as follows:

$$w_{\mathcal{X}}(y_i) = 2\left(1 - \frac{1}{1+\exp(-H_{\mathcal{X}}(y_i))}\right) \tag{2.4}$$

This reverse sigmoid is a decreasing function ranging between (0, 2). In practice, because the entropies are all positive, the weight coefficients range between 0 and 1. The weighted holo-entropy is defined as follows:

Definition 8. (Weighted Holo-entropy of a Random Vector) *The weighted holo-entropy $\mathcal{W}_{\mathcal{X}}(\mathcal{Y})$ is the sum of the weighted entropy on each attribute of the random vector $\mathcal{Y}$.*

$$\mathcal{W}_{\mathcal{X}}(\mathcal{Y}) = \sum\nolimits_{i=1}^{m} w_{\mathcal{X}}(y_i) H_{\mathcal{X}}(y_i) \tag{2.5}$$

The weighted holo-entropy is bounded according to the following proposition:

Proposition 2. $0 \leq \mathcal{W}_{\mathcal{X}}(\mathcal{Y}) \leq \frac{2m}{n+1}\log(n)$

Proof. Since $\frac{\partial[w_{\mathcal{X}}(y_i)H_{\mathcal{X}}(y_i)]}{\partial H_{\mathcal{X}}(y_i)} = \left[\frac{\exp(-H_{\mathcal{X}}(y_i))}{1+\exp(-H_{\mathcal{X}}(y_i))}\right]^2 > 0$, $\mathcal{W}_{\mathcal{X}}(\mathcal{Y})$ of each attribute is monotonically increasing with the attribute weight. When $H_{\mathcal{X}}(y_i) = 0$, the minimum $w_{\mathcal{X}}(y_i)H_{\mathcal{X}}(y_i) = 0$. When $H_{\mathcal{X}}(y_i) = \log(n)$, the maximum value is $\frac{2}{n+1}\log(n)$. Since $HL_{\mathcal{X}}(\mathcal{Y}) = \sum_{i=1}^{m} H_{\mathcal{X}}(y_i)$, the inequalities hold. □

To illustrate the effectiveness of weighted holo-entropy as an outlier factor, let's look back at the example in Table 2.2. The $\mathcal{W}_{\mathcal{X}\setminus\{x_o\}}(\mathcal{Y})$ column, which is impacted more by attribute *Degree* than by attribute *Age*, indicates Case 4 is more likely to be an outlier than the Cases from 5 to 8. In Section 5, we provide extensive experimental results that show it is generally more advantageous to use attribute weighting in practical applications. In Section 4, we show that the attribute weighting in Eq. 2.5 can be efficiently handled within the detection process.

2.3.4 A Formal Definition of the Outlier Detection Problem

To formally define outliers, we need to describe the condition for judging how exceptional a subset of objects is. The following definition of outliers is based on the weighted holo-entropy, supposing that the number of the desired outliers o is given.

A set of o candidates is the best if its exclusion from the original data set $\mathcal{X}$ causes the greatest decrease in the weighted holo-entropy value, compared to all the other subsets of $\mathcal{X}$ of size o.

Definition 9. (Outliers) *Given a dataset $\mathcal{X}$ with n objects and the number o, a subset $Out(o)$ is defined as the set of outliers if it minimizes $J_{\mathcal{X}}(\mathcal{Y}, o)$, defined as the weighted holo-entropy of $\mathcal{X}$ with o objects removed.*

$$J_{\mathcal{X}}(\mathcal{Y}, o) = \mathcal{W}_{\mathcal{X}\backslash Set(o)}(\mathcal{Y}) \tag{2.6}$$

where $Set(o)$ is any subset of o objects from $\mathcal{X}$. In other words

$$Out(o) = argmin\ J_{\mathcal{X}}(\mathcal{Y}, o) \tag{2.7}$$

Hence, outlier detection is now formulated be stated as an optimization problem. For a given o, the number of possible candidate sets for the objective function is $C_n^o = \frac{n!}{o!(n-o)!}$, which is very high. Moreover, one might have to determine the optimal value of o, i.e., how many outliers a dataset really has. A possible theoretical approach to this problem is to search for a range of values of o and decide on an optimal value of o by optimizing a certain variational property of $J_{\mathcal{X}}(\mathcal{X}, o)$. We leave this as a future research direction. For now, we will focus on developing practical solutions to the optimization problem.

2.4 New Outlier Detection Algorithms

In this section, we propose two greedy algorithms to solve the above optimization problem for outlier detection. Our algorithms are built upon several important properties of the holo-entropy. In the following discussion, we first show how the holo-entropy can be efficiently estimated when only one object is removed from the data set. This can be done using the information of the removed object, without the need of estimating the probability distribution of each attribute. In addition, we propose a method to estimate the upper-bound number and the candidate set of outliers to further reduce the search space for the optimization problem. Finally, we present the two algorithms accompanied with a complexity analysis.

2.4.1 A New Concept of the Outlier Factor

In addition to the high computational complexity of searching for the optimal subset, solving Eq. 2.7 also involves the problem of repeatedly estimating the weighted holo-entropy, which in turn requires estimation of probability distribution of each attribute. Thus, Eq. 2.7 is considered as a theoretical model of outliers for which approximate solutions need to be found. Interestingly, the difference in weighted holo-entropy can be estimated, especially when only one object is removed, without having to estimate attribute probabilities. This opens up the possibility of an efficient heuristic approach to solving Eq. 2.7.

Definition 10. (Differential Holo-entropy) *Given an object x_o of $\mathcal{X}$, the difference of weighted holo-entropy $h_{\mathcal{X}}(x_o)$ between the dataset $\mathcal{X}$ and the dataset $\mathcal{X}\backslash\{x_o\}$ is defined as the differential holo-entropy of the object x_o.*

$$\begin{aligned} h_{\mathcal{X}}(x_o) &= \mathcal{W}_{\mathcal{X}}(\mathcal{Y}) - \mathcal{W}_{\mathcal{X}\backslash\{x_o\}}(\mathcal{Y}) \\ &= \sum\nolimits_{i=1}^{m} \left[w_{\mathcal{X}}(y_i)H_{\mathcal{X}}(y_i) - w_{\mathcal{X}\backslash\{x_o\}}(y_i)H_{\mathcal{X}\backslash\{x_o\}}(y_i)\right] \end{aligned} \tag{2.8}$$

Since $w_{\mathcal{X}}(y_i)$ is defined as a reverse sigmoid function of the entropy $H_{\mathcal{X}}(y_i)$, the difference between $w_{\mathcal{X}}(y_i)$ and $w_{\mathcal{X}\backslash\{x_o\}}(y_i)$ is significantly smaller than the entropy $H_{\mathcal{X}}(y_i)$. So we simplify the differential holo-entropy using the following expression:

$$\hat{h}_{\mathcal{X}}(x_o) = \sum\nolimits_{i=1}^{m} w_{\mathcal{X}}(y_i) \left[H_{\mathcal{X}}(y_i) - H_{\mathcal{X}\backslash\{x_o\}}(y_i)\right] \tag{2.9}$$

Our preliminary experiment indicates that the performance of exact and approximate outlier factor are very similar. To avoiding the high time complexity of exact factor computation, we use the approximate factor to represent the approximate one in this work. The approximate differential holo-entropy $\hat{h}_{\mathcal{X}}(x_o)$ can be directly computed according to the following proposition:

Proposition 3. *The approximate differential holo-entropy $\hat{h}_{\mathcal{X}}(x_o)$ can be represented as follows:*

$$
\begin{aligned}
\hat{h}_{\mathcal{X}}(x_o) = & \sum_{i=1}^{m} w_{\mathcal{X}}(y_i)\left(\log a - \frac{a}{b}\log b\right) - aW_{\mathcal{X}}(\mathcal{Y}) \\
& + a\sum\nolimits_{i=1}^{m} \begin{cases} 0, \; if \; n(x_{o,i}) = 1; \\ w_{\mathcal{X}}(y_i) \cdot \delta\left[n(x_{o,i})\right], \; else. \end{cases}
\end{aligned}
\tag{2.10}
$$

where $\delta(x) = (x-1)\log(x-1) - x\log x$, and $x_{o,i}$ means the value appears in the ith attribute of the object x_o. $n(x_{o,i})$ is the simplified form of $n(i, x_{o,i})$, which means the times $x_{o,i}$ appears in the ith attribute. b and a are reciprocal values of the cardinality of $\mathcal{X}$ and $\mathcal{X}\backslash\{x_o\}$.

Proof. $\hat{h}_{\mathcal{X}}(x_o) = \sum_{i=1}^{m} w_{\mathcal{X}}(y_i)[H_{\mathcal{X}}(y_i) - H_{\mathcal{X}\backslash\{x_o\}}(y_i)]$; when $n(x_{o,i}) = 1$, $H_{\mathcal{X}}(y_i) - H_{\mathcal{X}\backslash\{x_o\}}(y_i)$ is written as

$$
\begin{aligned}
& a\sum\nolimits_{j=1,j\neq o}^{n_i-1}\left[n(x_{j,i})\log n(x_{j,i}) + n(x_{j,i})\log a\right] \\
& - b\sum\nolimits_{j=1,j\neq o}^{n_i-1}\left[n(x_{j,i})\log n(x_{j,i}) + n(x_{j,i})\log b\right] - b\log b;
\end{aligned}
$$

when $n(x_{o,i}) > 1$, $H_{\mathcal{X}}(y_i) - H_{\mathcal{X}\backslash\{x_o\}}(y_i)$ is written as

$$
\begin{aligned}
& a\sum\nolimits_{j=1,j\neq o}^{n_i-1}\left[n(x_{j,i})\log n(x_{j,i}) + n(x_{j,i})\log a\right] \\
& - b\sum\nolimits_{j=1,j\neq o}^{n_i-1}\left[n(x_{j,i})\log n(x_{j,i}) + n(x_{j,i})\log b\right] - a\log a \\
& + (a\log a - b\log b)n(x_{o,i}) - b \cdot n(x_{o,i})\log n(x_{o,i}) \\
& + a\left[n(x_{o,i}) - 1\right]\log\left[n(x_{o,i}) - 1\right].
\end{aligned}
$$

Combining these two situations, the deduced form of $\hat{h}_{\mathcal{X}}(x_o)$ is expressed as follows:

$$
\begin{aligned}
& \hat{h}_{\mathcal{X}}(x_o) = a\sum\nolimits_{i=1}^{m} w_{\mathcal{X}}(y_i) \cdot \delta\left[n(x_{o,i})\right] + \\
& \sum_{i=1}^{m}\left[\log\frac{a}{b} + ab \cdot \log\left(\frac{n(x_{1,i})}{n}^{\frac{n(x_{1,i})}{n}} \ldots \frac{n(x_{n_{i,i}})}{n}^{\frac{n(x_{n_{i,i}})}{n}}\right)\right]
\end{aligned}
$$

Since $\log\left(\frac{n(x_{1,i})}{n}^{\frac{n(x_{1,i})}{n}}\ldots\frac{n(x_{n_{i},i})}{n}^{\frac{n(x_{n_{i},i})}{n}}\right) = -\frac{1}{b}(E(y_i)+\log b)$, the simplified deduced form is

$$\hat{h}(x_o) = \sum\nolimits_{i=1}^{m} w_{\mathcal{X}}(y_i)\left(\log a - \frac{a}{b}\log b\right) - aW_{\mathcal{X}}(\mathcal{Y})$$
$$+\, a\sum\nolimits_{i=1}^{m}\begin{cases} 0, & if\ n(x_{o,i}) = 1; \\ w_{\mathcal{X}}(y_i)\cdot\delta\left[n(x_{o,i})\right], & else. \end{cases}$$

□

If we consider only the unweighted holo-entropy, i.e., all the attribute weights are treated as 1, Proposition 3 holds for the differential holo-entropy $h_{\mathcal{X}}(x_o)$. We will use this exact equation to derive the formula for updating entropiés and attribute weights in the next section. Also, according to Proposition 3, $\hat{h}(x_o)$ is determined by the dataset $\mathcal{X}$, i.e., in the first two terms, $\sum_{i=1}^{m} w_{\mathcal{X}}(y_i)\left(\log a - \frac{a}{b}\log b\right) - aW_{\mathcal{X}}(\mathcal{Y})$, and by the object x_o itself in the third terms. Based on these discussions, we define the outlier factor of an object as follows:

Definition 11. (Outlier Factor of an Object) *The outlier factor of an object x_o, denoted as $OF(x_o)$, is defined as*

$$OF(x_o) = \sum_{i=1}^{m} OF(x_{o,i}) = \sum_{i=1}^{m}\begin{cases} 0, & if\ n(x_{o,i}) = 1; \\ w_{\mathcal{X}}(y_i)\cdot\delta\left[n(x_{o,i})\right], & else. \end{cases} .$$

where $OF(x_{o,i})$ is defined as the outlier factor of x_o on the ith attribute.

$OF(x_o)$ can be considered as a measure of how likely it is that object x_o is an outlier. An object x_o with a large outlier factor value is more likely to be an outlier than an object with a small value. Here are a few other interesting properties of the outlier factor:

Proposition 4.
$OF(x_{u,i}) \geq OF(x_{j,i})$, if $n(x_{u,i}) = 1$ and $n(x_{j,i}) \geq 1$

Proof. The outlier factor has a negative or zero value on an attribute; when $x_{u,i}$ is unique, the outlier factor achieves its largest value, zero. So the proposition holds. □

Proposition 5.

$OF(x_{j,i}) \geq OF(x_{k,i})$, if $n(x_{j,i}) \leq n(x_{k,i})$ and $n(x_{j,i}) > 1$

Proof. Set $\alpha(x_{j,i}) = \left[\frac{n(x_{j,i})^{n(x_{j,i})}}{(n(x_{j,i})-1)^{n(x_{j,i})-1}}\right]^{w_{\mathcal{X}}(y_i)}$, $\varphi(x_{j,i}, x_{k,i}) = \frac{\alpha(x_{k,i})}{\alpha(x_{j,i})}$ and $\phi(x_{j,i}, x_{k,i}) = \log(\varphi(x_{j,i}, x_{k,i})) = OF(x_{j,i}) - OF(x_{k,i})$.

Since $\alpha'(x_{j,i}) = w_{\mathcal{X}}(y_i)\frac{{x_{j,i}}^{x_{j,i}}[Inx_{j,i} - In(x_{j,i}-1)]}{(x_{j,i}-1)^{x_{j,i}-1}}\left[\frac{n(x_{j,i})^{n(x_{j,i})}}{(n(x_{j,i})-1)^{n(x_{j,i})\ 1}}\right]^{w_{\mathcal{X}}(y_i)-1} > 0$, $\alpha(x_{j,i}) > 0$, hence $\varphi(x_{j,i}, x_{k,i}) \geq 1$, and thus $\phi(x_{j,i}, x_{k,i}) \geq 0$. When $n(x_{j,i}) = n(x_{k,i})$, the equality holds. □

According to propositions 4 and 5, for each attribute, the outlier factor is monotonically decreasing w.r.t. the frequency of the object value on that attribute. This corresponds to the following intuitive idea: given an object, regardless of the weight of an attribute, the higher the frequency of the object value on that attribute, the less likely it is that the object is an outlier.

2.4.2 Updating the Outlier Factor

In this section, we discuss the issue of updating the outlier factor within a constant time in a step-by-step process. To update $OF(x_o)$, according to Def. 11 and the definition of attribute weight in Eq. 2.4, we should first update the entropy of each attribute. Since the attribute entropy is always changing when outliers are detected and removed from the data set, the direct computation of $H_{\mathcal{X}\setminus\{x_o\}}(y_i)$ is very time-consuming. By a line of reasoning similar to the proof of Proposition 3, the unweighted differential holo-entropy $HL_{\mathcal{X}}(\mathcal{Y}) - HL_{\mathcal{X}\setminus\{x_o\}}(\mathcal{Y})$ can be deduced as follows:

$$\begin{aligned}&HL_{\mathcal{X}}(\mathcal{Y}) - HL_{\mathcal{X}\setminus\{x_o\}}(\mathcal{Y})\\ &= m\left[\left(\frac{a}{b} - a\right)\log a - (b+1)\log b\right] - bHL_{\mathcal{X}}(\mathcal{Y}) + a\sum\nolimits_{i=1}^{m}\begin{cases}0, & if\ n(x_{o,i}) = 1;\\ \delta\left[n(x_{o,i})\right], & else.\end{cases}\end{aligned} \tag{2.11}$$

Based on this expression, we can obtain the simple updated form of the holo-entropy $HL_{\mathcal{X}\setminus\{x_o\}}(\mathcal{Y})$ as:

$$HL_{\mathcal{X}\setminus\{x_o\}}(\mathcal{Y})$$
$$= (1+b)HL_{\mathcal{X}}(\mathcal{Y}) - m\left[\left(\frac{a}{b} - a\right)\log a - (b+1)\log b\right] - a\sum\nolimits_{i=1}^{m} \begin{cases} 0, & if\ n(x_{o,i}) = 1; \\ \delta\left[n(x_{o,i})\right], & else. \end{cases}$$

From this, the formula for each individual attribute entropy $H_{\mathcal{X}\setminus\{x_o\}}(y_i)$ is obtained:

$$H_{\mathcal{X}\setminus\{x_o\}}(y_i)$$
$$= (1+b)H_{\mathcal{X}}(y_i) - \left[\left(\frac{a}{b} - a\right)\log a - (b+1)\log b\right] - a\begin{cases} 0, & if\ n(x_{o,i}) = 1; \\ \delta\left[n(x_{o,i})\right], & else. \end{cases} \tag{2.12}$$

This can be efficiently implemented in a step-by-step process. After calculating the entropy by Eq. 2.12, we can easily compute the updated attribute weight using Eq. 2.4. Finally, using Def. 11, the outlier factor can be efficiently updated.

2.4.3 Upper Bound on Outliers

In unsupervised outlier detection, the majority of objects in a dataset are supposed to be normal objects [22]. How can we estimate an upper limit on the number of outliers in a dataset? And how can we divide the dataset into normal objects and anomaly (outlier) candidates? In this subsection, we introduce three new concepts: the upper bound on outliers (**UO**), the anomaly candidate set (**AS**) and the normal object set (**NS**).

These concepts are constructed on the assumption that eliminating outliers will improve the purity of the dataset and that this process reduces $\mathcal{W}_{\mathcal{X}}(\mathcal{Y})$. When a normal object is removed from the dataset, the value of $\mathcal{W}_{\mathcal{X}}(\mathcal{Y})$ should increase. Thus the objects with positive $\hat{h}(x_i)$ are defined as the anomaly candidate set (AS), $AS = \{x_i, |\hat{h}(x_i) > 0|\}$. The objects with non-positive $\hat{h}(x_i)$ are defined as elements of the normal object set (NS), $NS = \{x_i, |\hat{h}(x_i) \leq 0|\}$. The number of objects in AS

is defined as UO.

$$AS = \{x_i, |\hat{h}(x_i) > 0|\}$$
$$UO = N(AS) = \sum\nolimits_{i=1}^{n} \left(\hat{h}(x_i) > 0\right) \tag{2.13}$$

AS will be used as the outlier candidate set; i.e., only the UO objects from AS will be examined by our algorithms. For instance, the UO in Fig. 2.1(a) is 2, the AS contains two elements $\{a_1,\ b_2\}$ and $\{b_1,\ c_2\}$, and the rest of the objects $\{a_1, a_2\}$ are normal objects. Later in the paper, we will provide extensive evidence on the adequacy of limiting the outlier search to AS. It is worth pointing out that the normal object set NS can be of great interest as the candidate set for frequent-itemset mining and class-profile building. In this paper, we are focusing only on the use of AS for outlier detection. For the experimental datasets, the UO values are listed in Table 2.5. Note that the average UO is about $0.21n$.

2.4.4 ITB-SP and ITB-SS Algorithms

In this subsection, we make use of the outlier factor defined in Subsection 2.4.1 to derive two greedy algorithms for outlier detection. One is named **ITB-SS** for Information-Theory-Based Step-by-Step (or **SS** for short), the other one is named **ITB-SP** for Information-Theory-Based Single-Pass (or **SP** for short). Both algorithms detect outliers one by one. At each step of SS, the object with the largest $OF(x_o)$ is identified as an outlier and is removed from the dataset. Following this removal, the outlier factor $OF(x)$ is updated for all the remaining objects. The process repeats until o objects have been removed. In SP, the outlier factors are computed only once, and the o objects with the largest $OF(x)$ values are identified as outliers. In both algorithms, search is conducted only within the anomaly candidate set AS, although this does not make any difference for the algorithm ITB-SP since the initialization of AS requires computation of the outlier factors of all the objects. ITB-SS does benefit, however, from the reduced search space. In designing the two algorithms, we assumed that the number of requested outliers o is always smaller than UO. Experimental results in the next section show that AS is indeed large enough to include all the candidate objects that can reasonably be considered as outliers. Nevertheless, only minor modifications need to be made if a user wants to obtain more than UO "outliers".

Algorithm 1 : ITB-SP single pass

```
Input: dataset $\mathcal{X}$ and number of outliers requested $o$
Output: outlier set $OS$
Compute $w_{\mathcal{X}}(y_i)$ for $(1 \leq i \leq m)$ by Eq. 2.4
Set $OS = \phi$
for $i = 1$ to $n$ do
    Compute $OF(x_i)$ and obtain $AS$ by Eq. 2.13
end for
if $o > UO$ then
    $o = UO$
else
    Build $OS$ by searching for the $o$ objects with greatest $OF(x_i)$ in $AS$ using heapsort
end if
```

Let's look at the time complexity of ITB-SP (Algorithm 1). In ITB-SP, the attribute weights $w_{\mathcal{X}}(y_i)(1 \leq i \leq m)$, the $OF(x_i)$ of all the objects, initialization of AS and the heapsort search to find the top-o outlier candidates are computed. The time complexity of computing $w_{\mathcal{X}}(y_i)$ and $OF(x_i)$, including initialization of AS, is $O(mn)$, and the time cost of top-o searching is $O(nlog(o))$. Since the value of $log(o)$ is always much smaller than the number of attributes m in real applications, the final time complexity of ITB-SP can be written as $\mathbf{O(nm)}$.

For ITB-SS (Algorithm 2), the attribute weights, initial outlier factors including initialization of AS, and the step-by-step top-o outlier selection procedure are computed. The time cost of attribute weights, initial outlier factors and initialization of AS is $O(mn)$, and the time complexity of step-by-step top-o outlier selection from step 11-15 is $O(om(UO))$. Thus, the overall complexity is $O(nm + om(UO))$. Considering that $o(UO)$ is usually larger than n, it is possible to say that the final complexity of ITB-SS is $\mathbf{O(om(UO))}$. Compared with ITB-SP, the time complexity of the ITB-SS method is a little higher.

Algorithm 2 ITB-SS Step-by-Step

```
**Input**: dataset $\mathcal{X}$ and number of outliers requested $o$
**Output**: outlier set $OS$
Set $OS = \phi$ .
Compute $w_{\mathcal{X}}(y_i)$ for $(1 \leq i \leq m)$ by Eq. 2.4
for $i = 1$ to $n$ do
    Compute $OF(x_i)$ and obtain $AS$ by Eq. 2.13
end for
if $o > UO$ then
    $o = UO$
else
    for $i = 1$ to $o$ do
        Search for the object with greatest $OF(x_o)$ from $AS$
        Add $x_0$ to $OS$ and remove it from $AS$
        Update all the $OF(x)$ of $AS$
    end for
end if
```

2.5 Experiments

In this section, we conduct effectiveness and efficiency tests to analyze the performance of the proposed methods. To test effectiveness, we compare ITB-SS and ITB-SP with competing methods on synthetic and real datasets. For the efficiency test, we conduct evaluations on synthetic datasets to show how running time increases with the number of objects, the number of attributes and the number of outliers.

Table 2.3: Comparison among ITB-SP, ITB-SS and Optimal Solutions on Soybean Data

o	ITB-SP	$J_{\mathcal{X}}(\mathcal{Y}, o)$	ITB-SS	$J_{\mathcal{X}}(\mathcal{Y}, o)$	Optimal	$J_{\mathcal{X}}(\mathcal{Y}, o)$
1:	11	9.686	11	9.686	11	9.686
2:	11,18	9.687	11.18	9.687	11,18	9.687
3:	11,**15**,18	**9.687**	11,**15**,18	**9.687**	11,16,18	9.676
4:	11,15,16,18	9.671	11,15,16,18	9.671	11,15,16,18	9.671
5:	11,15,16,18,20	9.659	11,15,16,18,20	9.659	11,15,16,18,20	9.659
6:	11,15,**16**,18,19,20	**9.646**	11,13,15,18,19,20	9.642	11.13,15,18,19,20	9.642
7:	11,13,15,16,18,19,20	9.585	11,13,15,16,18,19,20	9.585	11,13,15,16,18,19,20	9.585
8:	11,13,**14**,15,16,18,19,20	**9.541**	11,13,15,16,17,18,19,20	9.537	11,13,15,16,17,18,19,20	9.537
9:	11,13,14,15,16,18,19,20,**29**	**9.493**	11,13,14,15,16,17,18,19,20	9.468	11,13,14,15,16,17,18,19,20	9.468
10:	11,12,13,14,15,16,18,19,20,**29**	**9.419**	11,12,13,14,15,16,17,18,19,20	9.334	11,12,13,14,15,16,17,18,19,20	9.334

2.5.1 Compared Methods and Experiment Outline

For our experiments, we implement and compare our algorithms with several mainstream methods for categorical outlier detection. These representative methods include CNB from the proximity-based approach and FIB and OA from the rule-based approach. Since the anomaly candidate set (AS) is utilized as a pruning facility to reduce the time complexity of the proposed methods, ITB-SS and ITB-SP can be considered as top-N outlier detection methods [67]. To the best of our knowledge, for categorical outlier detection, there is no other clear claim in the literature of a top-N outlier detection method. Some efficient top-N methods do exist for numerical outlier detection [59][60], but these methods cannot be easily adapted to deal with categorical data because to reach the top-N they explore properties of their distance measures that are difficult to generalize to categorical data. In a preliminary test, we tried to adapt the LOF method [17] and its efficient top-N variation [59] with a micro-cluster pruning mechanism [60] to categorical data sets. The adapted methods did not work very well in our experiments. For reasons of fairness, we decided not to include any comparison with an adapted method from numerical outlier detection.

Various experimental results are reported in this section. To evaluate the proposed methods, we begin by comparing the performance of ITB-SS and ITB-SP with the optimal solutions obtained by exhaustive search on a small real data set. Although limited in the size of the test data set, this experiment illustrates that the proposed methods are able to provide very good solutions to the high-complexity optimization problem. Experiments on different synthetic data in this section can be used as evidence to illustrate the effectiveness and stability of the proposed methods for large-scale datasets. Outlier factors of different methods are compared to gain a better understanding of the advantage of the proposed methods. Extensive comparisons on real data sets allow us to judge the effectiveness of the proposed methods in comparison with other methods. Moreover, we include in these comparisons the detection performance of ITB-SS and ITB-SP in both their weighted and unweighted versions. This illustrates the benefit and importance of weighting the attributes. Finally, to evaluate the efficiency of the proposed methods, synthetic datasets are utilized to test the run time w.r.t. increasing numbers of objects, attributes and outliers.

2.5.2 Effectiveness Test

Evaluation of Approximation

This subsection reports on experiments conducted to see whether the solutions obtained by ITB-SS and ITB-SP are close to the optimal solutions obtained by optimizing the object function $J_{\mathcal{X}}(\mathcal{Y}, o)$. The dataset used is the public, categorical "soybean data" [3], with 47 objects and 35 attributes. This data contains a very small class of 10 objects (numbers 11 to 20 in the original dataset). Since the data does not have explicitly identified outliers, it is natural to treat the objects of the smallest class as "outliers". Therefore, we should check whether objects from this class will be detected for $o = 1, ..., 10$.

Table 2.3 shows different sets of "outliers" obtained by ITB-SP, ITB-SS, and the optima for different values of o. The $J_{\mathcal{X}}(\mathcal{Y}, o)$ values in **bold-faced** letters indicate the cases where non-optimal sets were detected by either ITB-SP or ITB-SS, while the subsets of objects 11 to 20, which originally belong to the smallest class, found by strictly optimizing the $J_{\mathcal{X}}(\mathcal{Y}, o)$ are taken as reference sets of optimality. It can be observed that ITB-SS seems to be quite effective, since it falsely detects an outlier subset only once in the ten tries. As can be anticipated, ITB-SP makes more mistakes (five out of ten subsets). Nevertheless, the ITB-SP process is able to approximate the optimal solutions quite well when more and more outliers are detected. Also, if we look at the outlier output of each detection step, there is never more than one wrongly detected object. Similar phenomena have been observed with our other evaluations of approximation experiments.

Test of Outlier Factors

The experiments reported in this subsection help to understand why ITB-SS and ITB-SP are effective in solving the outlier detection problem. Here, we show some important differences between the outlier factors used in different algorithms. For this purpose, we make use of a synthetic dataset, illustrated in Table 2.4 by $y_1, ..., y_8$, and compare the outlier factor values, also illustrated in Table 2.4. The 13 objects are different from each other. In order to visualize the dataset, we draw a two-dimensional representation in Fig. 2.2, using the principle of graph drawing [31]. In this graph, the vertices indicate the objects and the edges represent the similarity between objects, where all the similarities are 1. The columns CNB, FIB, OA and ITB show the outlier factor values of each object obtained by the compared methods. Note that for OA, CNB and ITB, an object with a larger outlier factor is more likely to be an outlier, while for FIB the opposite is true. The column ITB represents $OF(x_o)$ defined in this paper. The settings of the parameters for the other methods, are as follows: similarity threshold and number of nearest neighbors in CNB are set to $\theta = 0.1$ and $k = 2$; minimum support rate in OA and FIB is set to $\mathit{SupRate} = 0.1$.

Table 2.4: Outlier Factors on a Synthetic Dataset

obj.	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	CNB	FIB	OA	ITB
1:	k_1	j_2	k_3	k_4	e_5	c_6	e_7	b_8	2.00	0.44	25.0	-0.51
2:	j_1	i_2	j_3	j_4	b_5	c_6	d_7	e_8	2.00	0.44	25.0	-0.51
3:	i_1	h_2	i_3	i_4	d_5	d_6	b_7	a_8	2.00	0.44	25.0	-0.48
4:	h_1	g_2	h_3	h_4	c_5	b_6	b_7	d_8	2.00	0.44	25.0	-0.51
5:	g_1	f_2	g_3	g_4	b_5	a_6	a_7	b_8	2.00	0.89	23.0	-0.99
6:	f_1	e_2	f_3	f_4	a_5	b_6	a_7	a_8	2.00	0.89	23.0	-0.99
7:	a_1	a_2	c_3	c_4	a_5	a_6	c_7	c_8	2.00	0.89	23.0	-1.01
8:	a_1	b_2	a_3	a_4	f_5	e_6	f_7	f_8	2.00	0.89	23.0	-0.99
9:	b_1	a_2	a_3	b_4	g_5	f_6	g_7	g_8	2.00	0.89	23.0	-0.99
10:	c_1	b_2	b_3	d_4	h_5	g_6	h_7	h_8	2.00	0.44	25.0	-0.51
11:	d_1	d_2	b_3	a_4	i_5	h_6	i_7	i_8	2.00	0.44	25.0	-0.48
12:	b_1	c_2	d_3	e_4	g_5	i_6	j_7	j_8	2.00	0.44	25.0	-0.51
13:	e_1	c_2	e_3	b_4	k_5	g_6	k_7	k_8	2.00	0.44	25.0	-0.51

The results indicate that our proposed factor $OF(x_o)$ for ITB better reflects the intuitive understanding of the dataset. Specifically, the column CNB shows that all objects obtain the same outlier factor value. So for CNB, all the objects are equally likely to be outliers. FIB and OA make a similar distinction between objects 5-9 and the rest of the objects. They improve on the assessment of CNB by assigning a greater likelihood of being outliers to objects 1-4 and 10-13. It is ITB that provides

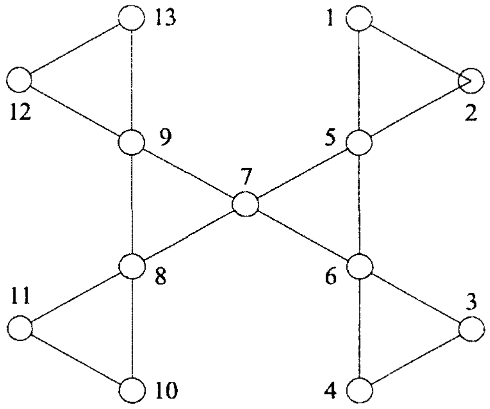


Figure 2.2: Graph Drawing of the Synthetic Dataset

the most precise assessment. It indicates that object 7 in the middle of the dataset is less likely to be an outlier than objects 5, 6, 8 and 9, which are similar to each other but have a common similar object 7. Moreover, objects 5, 6, 8 and 9 are less likely to be outliers than objects 1-4 and 10-13, each of which is similar to only two other objects. These differences are important indices used by ITB-SP and ITB-SS to accurately identify the most likely outlier candidate.

Test on Real Datasets

A large number of public real datasets, most of them from UCI [3], are used in our experiments, representing a wide range of domains in science and the humanities. Some of them have already been used as benchmarks for intrusion and outlier detection [92][49][75]. Some datasets such as web-advertisement [3][2] and sampled KDD Cup 1999 Data [3][3] contain already labelled anomaly objects. The others are categorical or mixed-type datasets with class labels representing many different data distributions in the real world. For these data, we use the same strategy as [49][75] to choose the

[2] The web-advertisement data represents a snapshot of image advertisements that have appeared on Internet pages. It is composed of major objects of 'normal' images and some 'bad' images, i.e. advertisements.

[3] The 10-percent KDD Cup 1999 Data has some attacks and 'good' normal connections. Since the number of attacks is greater than the number of normal connections, we select a total of 157663 normal objects and randomly choose 11213 attacks to make the 'bad' objects occupy a small part of the whole dataset.

objects in the smallest classes as the most likely anomalies.

Numeric attributes in these real data sets are, for the sake of simplicity, discretized by 10-bin discretization [33]. It is possible to adapt ITB-SS and ITB-SP to continuous attributes either through extending the holo-entropy, or through a more sophisticated discretization method [33], e.g., equal distance discretization, equal frequency discretization, unsupervised clustering methods and so on. But this may require an extensive effort and will be investigated as part of our future work. For the experiments in this paper, the adopted discretization scheme is fair for all the tested algorithms.

The other general setting of our experiments is as follows: All the missing values are replaced with the modes in the corresponding categorical attributes. The Area Under the Curve (**AUC**) (curve of detection rate and false alarm rate) [22][52] and significance test are used to measure the performance. The AUC results of different methods and the characteristics of all test datasets, such as the numbers of objects (#n), attributes (#m) and outliers (#o) and the upper bound on outliers (#UO), are summarized in the upper part of Table 2.5. There is no result for CNB on the KDD dataset because the time and space complexities of CNB are too high for this large set. Similarly, there is no result for either FIB or OA on the web advertisement data set, because the dimensionality of this set is too large for FIB and OA. The **bold-faced** AUC indicates the best method(s) for a particular data set. The parameters in the compared algorithms are set as suggested, i.e., $\theta = 0.3$, $k = 5$ in CNB and $SupRate = 0.3$, $MaxItem = 5$ in FIB and OA.

The results reported in Table 2.5 warrant a number of comments. First, between the weighted and unweighted versions of the proposed methods, the results in the last four columns of Table 2.5 show that the performance of the weighted version generally surpasses that of the unweighted version. These results are evidence of the importance of capturing attribute weights. Moreover, the Average line indicates that the improvement of ITB-SS over unweighted ITB-SS is much more significant than the improvement of ITB-SP over unweighted ITB-SP. This difference can be explained by the repeated weight updating in the ITB-SS method each time an outlier is detected and removed, whereas ITB-SP does not involve weight updating. We remark that

Table 2.5: AUC Results of Tested Algorithms on the Real and Synthetic Datasets

	Dataset	#n	#m	#o	#UO	CNB	FIB	OA	unweighted ITB-SP	ITB-SP	unweighted ITB-SS	ITB-SS
Real Data-sets	autos	133	26	12	58	0.588	0.753	0.588	**0.786**	0.762	0.776	0.757
	breast-c.	495	11	45	125	0.993	0.909	**0.996**	0.991	0.993	0.993	**0.996**
	breast-w.	699	10	241	281	0.975	0.989	0.989	0.984	0.985	0.990	**0.992**
	credit-a	413	17	30	171	0.844	0.926	0.875	0.888	0.935	0.925	**0.969**
	diabetes	768	9	268	340	0.869	0.885	0.769	0.758	0.797	0.835	**0.907**
	ecoli	336	8	9	144	0.894	0.921	0.965	0.968	0.986	0.974	**0.989**
	glass	187	10	12	83	0.566	0.681	0.681	**0.782**	0.767	0.773	0.748
	heart-h	294	14	106	132	0.650	0.780	0.695	0.727	0.728	**0.842**	0.800
	heart-s.	270	15	120	128	0.707	0.778	0.788	0.705	0.707	0.827	**0.849**
	hepati.	155	21	32	72	0.714	0.870	0.876	0.831	0.854	0.888	**0.903**
	ionosph.	351	45	126	183	0.559	0.492	0.563	0.554	0.614	0.561	**0.681**
	kr-vs-kp	1829	37	160	733	**1.000**	0.955	0.937	0.939	0.935	0.955	0.953
	labor	57	17	20	30	0.453	0.762	0.811	0.568	0.647	0.717	**0.873**
	splice	1795	61	140	897	0.568	0.878	0.635	0.995	**0.996**	**0.996**	**0.996**
	tic-tac-toe	688	10	62	294	0.996	**1.000**	0.966	**1.000**	0.967	**1.000**	0.955
	voting	293	17	26	101	**0.989**	0.976	**0.989**	0.966	0.974	0.977	0.984
	vowel	750	14	30	306	0.679	0.577	**1.000**	0.834	0.798	0.801	0.781
	zoo	90	18	6	53	0.300	**0.844**	0.597	0.784	0.746	0.831	0.816
	KDD	168876	42	11213	32923	*-	0.930	0.940	0.937	0.945	0.953	**0.954**
	police	122	3	7	18	0.882	0.988	0.977	0.981	**0.993**	**0.993**	**0.993**
	web-ad.	3279	1558	458	736	0.719	*-	*-	0.705	0.701	0.735	**0.735**
Average Results of Real Datasets						0.747	0.845	0.832	0.842	**0.852**	0.873	**0.890**
Synth. Data-sets	Data1	1000	10	50	50	0.718	0.821	0.816	1.000	1.000	1.000	1.000
	Data2	5000	10	250	1438	0.773	0.793	0.771	1.000	1.000	1.000	1.000
	Data3	1000	100	50	172	0.638	0.781	0.653	0.998	1.000	1.000	1.000
	Data4	5000	100	250	1424	0.545	0.543	0.668	0.999	1.000	1.000	1.000

the unweighted ITB-SP and the unweighted ITB-SS do outperform their weighted counterparts occasionally. This may be caused by the way "outliers" are determined and by non-representative objects that do not allow reliable estimation of attribute weights.

Now, let us look at the comparison between our proposed methods and the compared methods. The results in Table 2.5 reveal that our proposed methods are more effective than CNB, FIB and OA. The table shows that ITB-SS outperforms these methods on more than 70% of all datasets. The Average row of the AUC value also indicates that ITB-SS performs much better overall than the other methods, followed by ITB-SP, FIB and OA. More importantly, ITB-SS is effective on the large dataset *KDD* and on the high-dimensional dataset *web-ad*.

In order to determine whether the differences in outlier detection accuracy are statistically significant, we perform a pairwise comparison. The results are presented in Table 2.6. Each cell in the table contains the number of datasets for which the method in the row, i.e., ITB-SP or ITB-SS, wins, loses or ties relative to the corresponding method in the column, over the selected 21 datasets. For detecting ties (statistically similar results), we use a two-tailed T-Test [32] with a significance level of 0.005. The pairwise comparison shows that ITB-SP and ITB-SS are more accurate than the other methods on these datasets. ITB-SS outperforms every other method in at least 13 datasets, and underperforms in at most 4 of them. ITB-SP, although not as effective as ITB-SS, outperforms the other compared methods on at least 11 data sets and loses on at most 7 data sets.

Table 2.6: Results of Significance Test (win/lose/tie)

	CNB	FIB	OA	ITB-SP	ITB-SS
ITB-SS	**18**/1/2	**16**/2/3	**17**/1/3	**13**/4/4	
ITB-SP	**14**/4/3	**11**/7/3	**11**/5/5		4/**13**/4

Test on Synthetic Datasets

We also compare the effectiveness of different methods on synthetic datasets in a relatively ideal setting, since the generated outliers are usually more distinctive than those in real data and the outliers "truth" can be used to verify whether an outlier algorithm is able to find them. Four experiments are reported in the bottom part of Table 2.5 [4], where the outliers take up 5% of the corresponding dataset. In fact, to generate each test set, the data generator [2] is first used to generate rule-based categorical datasets with ten clusters. Then 95% of the objects of the test set are obtained by randomly choosing from three of the ten generated clusters. These are considered to be normal objects. On the other hand, 5% of objects are randomly chosen from the remaining clusters and are considered to be outliers.

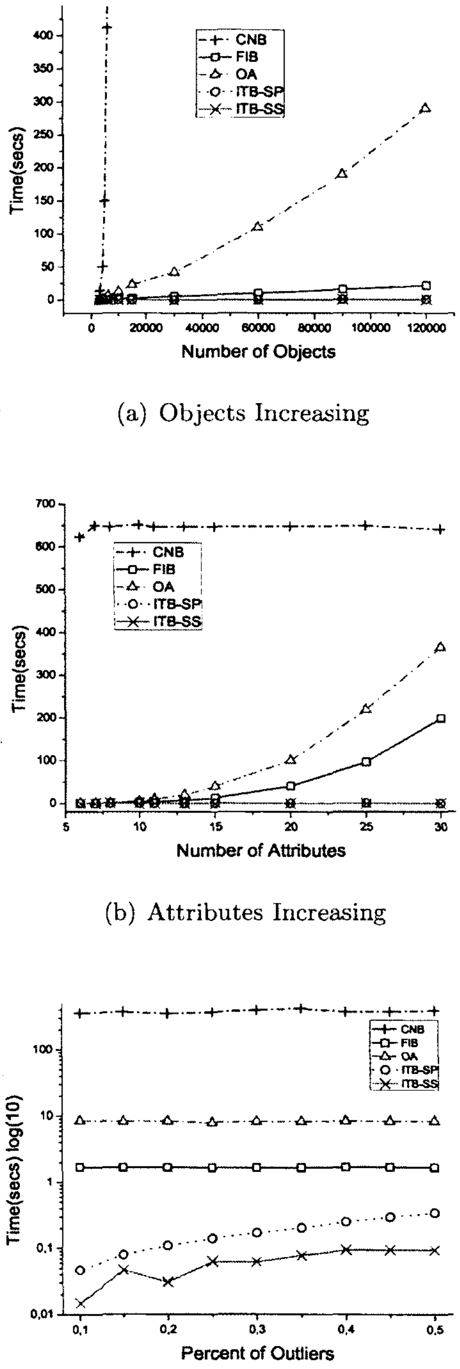
The results in Table 2.5 and in our other non-reported experiments show that synthetic datasets are in general too easy for ITB-SS and ITB-SP, as they often achieve near-perfect results. In general, these experiments confirm that the performance of CNB, FIB and OA is acceptable when the dimensionality of the data is not too high. Their performance declines quickly with an increasing number of dimensions. Increasing data size seems to hurt the performance of these methods too, but more extensive experiments are needed to draw a definitive conclusion.

2.5.3 Efficiency Test

To measure the time consumption with increasing numbers of objects, attributes and outliers, we employ GAClust [1] to generate synthetic datasets for these experiments. In the "objects increasing" test, the number of objects is increased from 3000 to 120000. In the "attributes increasing" test, the number of attributes increases from 6 to 30 [5]. In the "percentage of outliers increasing" test, we assume the percentage of outliers in a data set is increased from 10% to 50%. The results are shown in Fig. 2.3.

[4] Since FIB and OA have high time complexities with attributes and CNB is not able to deal with large datasets, we have set relatively small upper limits for the numbers of attributes and of objects, i.e. 100 and 5000 respectively. Our algorithm is effective to deal with large-scale datasets, e.g., the KDD data set with 168876 objects and the web advertisement dataset with 1558 attributes.

[5] To avoid the high time costs of FIB and OA, we set a relatively small upper limit on the number of attributes, i.e., 30 in this test.



(a) Objects Increasing

(b) Attributes Increasing

(c) Percent of Outliers Increasing

Figure 2.3: Results of Efficiency Test on Synthetic Datasets

All of the compared methods were implemented with C++, and run on a desktop with Intel Core 2 Quad processor (clocked at 2.4 GHz) and 4G memory.

As Fig. 3.3(a) indicates, the run times of ITB-SP, ITB-SS and FIB are almost linear functions of the number of objects. FIB has a higher increase rate than ITB-SP and ITB-SS. From the theoretical analysis, we know that the time complexity of CNB [75] increases quadratically with the number of objects, which is confirmed by the experimental data of Fig. 3.3(a). For the attributes increasing test, Fig. 3.3(b) shows that the run times of the FIB and OA increase rapidly with the number of attributes, which closely matches the theory that the time complexities of FIB [49] and OA [92] increase quadratically with the number of attributes. Compared with the time increase of FIB and OA, the increases for the other methods are too small to be noticeable on the figure. Fig. 2.3(c) illustrates the run time as a function of the percentage of "outliers" in the dataset each method is asked to search for. The time axis is in the log(10) scale. The run times of CNB, OA and FIB remain almost fixed with the "outlier percentage". Those of ITB-SP and ITB-SS methods increase linearly, but remain much lower than those of other methods even for very high "outlier percentages".

The three efficiency tests suggest ITB-SP and ITB-SS are efficient. They are particularly appropriate for large datasets with high dimensionality, and are also suitable for datasets with a high percentage of outliers. The CNB algorithm is not suitable for large datasets. The FIB and OA algorithms are not suitable for high-dimensional datasets, due to their high time complexities.

2.6 Conclusion

In this paper, we have formulated outlier detection as an optimization problem and proposed two practical, unsupervised, 1-parameter algorithms for detecting outliers in large-scale categorical datasets. The effectiveness of our algorithms results from a new concept of weighted holo-entropy that considers both the data distribution and attribute correlation to measure the likelihood of outlier candidates, while the efficiency of our algorithms results from the outlier factor function derived from the holo-entropy. The outlier factor of an object is solely determined by the object and its updating does not require estimating the data distribution. Based on this property, we apply the greedy approach to develop two efficient algorithms, ITB-SS and ITB-SP, that provide practical solutions to the optimization problem for outlier detection. We also estimate an upper bound for the number of outliers and an anomaly candidate set. This bound, obtained under a very reasonable hypothesis on the number of possible outliers, allows us to further reduce the search cost.

The proposed algorithms have been evaluated on real and synthetic datasets, and compared with different mainstream algorithms. First, our evaluations on a small real dataset and a bundle of synthetic datasets show that the proposed algorithms do tend to optimize the selection of candidates as outliers. Moreover, our experiments on real and synthetic datasets in comparison with other algorithms confirm the effectiveness and efficiency of the proposed algorithms in practice. In particular, we show that both of our algorithms can deal with datasets with a large number of objects and attributes.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions allowing to improve the quality of this paper. The work has been funded by Discovery and Discovery Accelerator Supplements programs of Natural Sciences and Engineering Research Council of Canada (NSERC) granted to Prof. Shengrui Wang. Shu Wu has been partly supported by the Ph.D. Scholarship program of the China Scholarship Council.

Chapter 3

Rating and Preference Framework for Highly Skewed User Rating Datasets

In this Chapter, we present a new general Bayesian rating and high-order preference framework RP for highly skewed rating datasets. This framework provides the probability can be used to create novel and efficient models to capture the users' high-order preference among items. This framework is composed by a generic optimization criterion **OptRP** and an efficient algorithm **LearnRP** to learn CF models w.r.t this criterion. As examples, we show how RP can be used to generate new and more effective matrix factorization models and adaptive neighborhood-based models. Experimental results on typical highly-skewed rating datasets show that the learned CF models provide significant improvements over the original ones, not only for rating prediction but also for *Top-N* recommendation.

This included paper has been submitted to *ACM Transactions on the Web*, June, 2012.

A Quantitative High-order Preference Framework for Highly Skewed User Rating Datasets

Shu Wu and Shengrui Wang[1]

Abstract

The collaborative filtering (CF) approach to rating prediction in recommender systems has received much attention recently. Most of the previous work has concentrated on movie or music rating datasets, and has enhanced the effectiveness of CF methods for rating prediction by using more complex expressions to represent the rating values or by importing additional information. However, little effort has been made to examine the rating behavior of users. For instance, when the distribution of ratings is highly skewed (i.e., the ratings are biased), existing methods become ineffective and unable to generate realistic ratings. Such skewed rating datasets are frequent in many real Web applications including product review sites such as Ciao and Epinions, video recommendation sites such as Youtube, and e-commerce product retailers such as Amazon and Ebay. In this paper, we extract high-order information on user preferences and employ it to enhance prediction quality for skewed rating datasets. We propose a new general framework for Rating and pairwise Preference (**RP**) comprising a generic optimization criterion **OptRP** and an efficient algorithm **LearnRP** to learn CF models w.r.t this criterion. As examples, we show how RP can be used to generate new, more effective matrix factorization models and adaptive neighborhood-based models. Experimental results on typical highly skewed rating datasets show that the learned CF models yield significant improvements over the original ones, not only for rating prediction but also for *Top-N* recommendation.

[1] The authors are with the Department of Computer Science, University of Sherbrooke, Sherbrooke, QC J1K 2R1, Canada. E-mail: {shu.wu, shengrui.wang}@usherbrooke.ca.

Keywords: Recommendation system, Collaborative filtering, High-order preference, Rating prediction, *Top-N* recommendation

Rating prediction has received considerable attention in recent years due to the advent of recommender systems on the Web to provide query, news article, tag, friend, location, next-basket and citation recommendations, among others. [115, 136, 74, 27, 81, 111, 103, 78, 96, 135, 122, 135, 100, 46]. Collaborative filtering (**CF**) [102] systems based on the opinions of users have become popular. CF for rating prediction can generate personalized item recommendations for users by collecting or gleaning information on their preferences based on past explicit or implicit behavior: e.g., ratings, transactions or even Web click streams. CF systems fall into two broad categories: those that use the neighborhood approach [30, 107, 77, 16, 125] and those that adopt the model-based approach [53, 105, 5, 104, 102, 65, 66, 66]. A good CF system will not only enhance customer satisfaction but also promote sales. Many e-commerce and Web service providers like Amazon [77] and Netflix (Cinematch) have adopted CF recommendation systems to further their businesses.

In a typical rating scenario, there are a set of users and a set of items. Each user is associated with a set containing all the items this user has rated. Each user-item rating value indicates the strength of the user's preference for the corresponding item. Typically, the range of ratings is in the bound $[1, 5]$. As mentioned in [61], previous work on rating prediction focused on one of two main problems: predicting whether or how much a particular user will like a particular item (called **rating prediction** in this work) and identifying a set of N items that will be interesting to a certain user (named ***top*-N recommendation**). The objective of the rating prediction task is to minimize the prediction error on all unknown ratings, while *top-N* recommendation places the emphasis on the predictive quality of personalized top items which should be more attractive for specific users.

3.0.1 User-specific Highly Skewed Rating Distribution.

Generally, in actual rating systems, there are two major kinds of rating scenarios, which we call **intent rating** and **emotional rating**. In previous work [44, 112], user rating motivations are explained using an economic model on the Movielens dataset. A user pays a cost for each rating in the form of mental effort or time, but benefits by receiving more accurate recommendations, keeping a list of products or having fun. According to the economic paradigm, users continue to provide ratings as long as they perceive that the benefits of a rating outweigh its costs [112]. In this work, we use "intent rating" to designate rating behavior towards a specific kind of items, aimed at benefiting by receiving more accurate recommendations. On the other hand, we name rating behavior intended to express opinions or influence others' choices "emotional rating" behavior.

The user-wise rating distributions generated by these two different rating behaviors usually possess different statistical properties. To get an intuitive idea about the rating distributions of intent and emotional rating datasets w.r.t. skewness measurements, let us look at Fig. 3.1, which presents the user-wise rating distributions and corresponding average skewness of six sets of rating data, the Netflix, Movielens 10M, Movielens 1M, Epinions, Amazon and Ciao datasets. The intent ratings presented in the first row of Fig. 3.1 are more symmetric and relatively evenly distributed on both sides of the mean in the average sense. On the other hand, the emotional rating distributions illustrated in the second row of Fig. 3.1 show higher asymmetry and higher skewness.

Most of the existing work on rating prediction has been focused on intent rating datasets. These datasets typically include movie and music rating datasets such as the EachMovie, MovieLens, Netflix and Yahoo Music datasets. The datasets in the first row of Fig. 3.1 have the majority of their ratings in the middle range, i.e., ratings 3 and 4. These ratings are distributed more symmetrically around the mean rating, compared with the emotional rating datasets. In fact, users tend to choose movies or songs belonging to genres matching their interests, and provide objective feedback on the items they have purchased and consumed, expecting the system to recommend the most appealing products to them. For instance, in the Netflix rating datasets, the ratings are provided by the user for movies he or she has watched. The user-specific rating distribution is likely to be more symmetric around the user's mean rating, and the average skewness of user-specific ratings tends to be small.

Emotional rating behaviors and emotional rating datasets are common on e-commerce sites, such as Ebay, Amazon and Taobao, product review websites, such as Epinions and Ciao, and content provider websites, such as Youtube. Generally, these emotional rating datasets can be categorized into two kinds of user behavior scenarios. On the e-commerce retailing and product review websites, users give the highest positive reviews to the most desirable products they have purchased. Before the act of purchase, the user may have compared this product with others. Therefore, the user likely chose the most satisfying one to buy. Thus this purchased product is more likely to be given the highest rating. In contrast to the intent rating datasets on movies and songs, the users in emotional rating datasets may spend much more money on these desired products. Previous research [24] on e-commerce retailing has also confirmed that consumers are more likely to rate products in the high rating scales. The second scenario of emotional rating behavior always exists in content provider websites. Users explore lots of items, such as videos, articles, jokes and images, and are likely to assign the most positive ratings to only a small portion of all the consumed items they find most appealing, to express their opinion or influence others' choices. In both of these scenarios, the user-specific rating distributions of these datasets are prone to show high skewness and long tails compared with those of the intent rating datasets.

The user-wise rating distributions for these emotional rating datasets, i.e., review ratings on Epinions and Ciao, and product ratings on Amazon, are shown in the second row of the Fig. 3.1. In contrast to the user-wise rating distributions of intent rating datasets, the majority of ratings in emotional rating datasets always fall under the highest rating value 5. The user-specific distributions of emotional rating datasets display higher skewness than those of intent rating datasets. These figures indicate that the distribution of Epinions has the highest skewness, followed by the Amazon and Ciao datasets.[2]

Other rating datasets also exhibit the distribution of emotional rating datasets. For instance, the rating datasets from three sites, eBay, Amazon and Yahoo, collected by the authors of [24], indicate that most users on these auction sites, both sellers and buyers, have mean ratings at the highest rating level and highly skewed rating

[2] The empirical results in Section 3.4 also confirm that there is a correlation between performance improvement and the skewness of user-specific rating distributions.

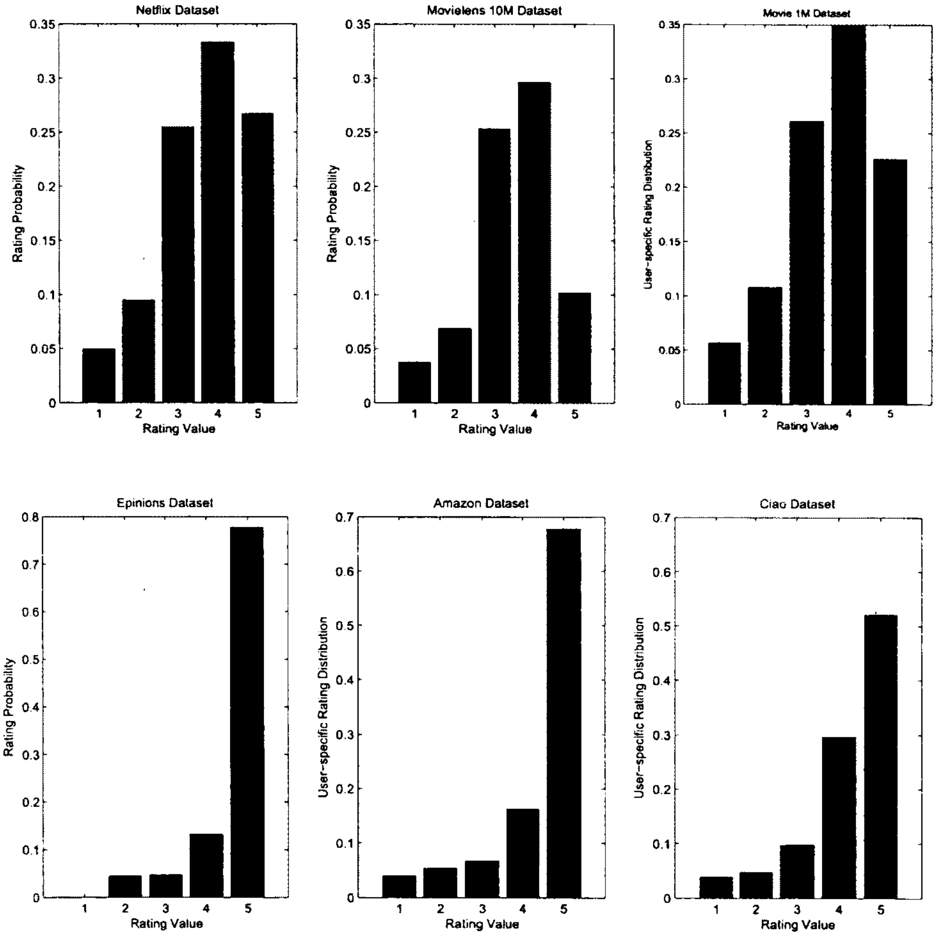


Figure 3.1: The average user-specific rating distributions of Netflix, Movielens 10M. Movielens 1M and Epinions, Amazon, Ciao datasets. The average skewness of these datasets is ($\mathcal{S}$=0.6023), ($\mathcal{S}$=0.5647), ($\mathcal{S}$=0.5536), ($\mathcal{S}$=1.7014), ($\mathcal{S}$=1.3387), ($\mathcal{S}$=0.7214) respectively.

distributions. Youtube also demonstrates this kind of rating behavior. Usually, users watch lots of videos on Youtube and rate only a small portion of them. Most of these ratings are toward the high end of the rating spectrum, as illustrated in the webpage[3]. Rating values of one, two, three and four only made up about 10% of observed ratings [84] and the marginal distribution showing the proportion of each rating in the observed data displays high skewness. This is also the reason why Youtube changed its 5-star rating scale to a like/dislike evaluation.

From the above observations, we notice that user-specific highly skewed rating datasets usually

- reveal users' preference towards items belonging to many different categories, e.g. electronics, computers, books, etc., rather than being limited to one category; or
- indicate users' preference w.r.t. a small portion of consumed content, e.g. videos, news and jokes, on content provider websites.

In either of these two situations, the users are more likely to give items the highest ratings. Therefore, the user-wise rating distributions of these datasets are prone to high skewness.

3.0.2 Objectives and contributions.

For the highly skewed rating datasets yielded by the action of emotional rating, we adopt high-order preference information to enhance predictive accuracy. As formally defined later in this paper, personalized high-order preference measures users' relative rating information across items, which we call users' quantitative preference between items. This high-order information is different from pointwise rating information (separate and independent user-item rating), which measures the rating strength on specific items. In conventional methods, pointwise rating information is captured by the conventional and widely adopted CF rating-based metrics, the Root Mean Square Error (**RMSE**) and Mean Absolute Error (**MAE**) [65, 66, 102, 51]. On highly skewed

[3] http://youtube-global.blogspot.com/2009/09/five-stars-dominate-ratings.html

user rating datasets, these conventional metrics are incapable of exactly capturing the user's high-order quantitative preference. This is because the prediction can achieve a good optimization of the rating-based metric to satisfy the majority ratings at one end of the rating range without paying much attention to the small proportion of ratings at the other end. On datasets with high user-specific skewness, the additional high-order preference constraints can capture the relative information among the ratings well and greatly improve predictive performance.

On highly skewed rating datasets, let us look at how high-order preference information is necessary to capture the relative preference of a user across rated items. Suppose that a user u rates items $\{i_1, i_2, i_3, i_4, i_5, ...\}$ with one lowest rating and many highest ratings $\{1, 5, 5, 5, 5, ...\}$. High-order preference information can reveal the degree to which u prefers i_1 least compared with all other items. Conventional pointwise ratings cannot precisely capture this relation. For example, in the matrix-factorization-based method[4], the predicted ratings may be averagely close to the given ratings, e.g. $\{4.9, 4.9, 5.2, 5.1, 4.8, ...\}$. Such rating predictions are considered good but reflect high-order preference poorly. For example, the predicted strength of preference for i_2 over i_1 is zero, which is totally different from the observed high-order preference, where u greatly prefers i_2 over i_1. For *top-N* recommendation, the item i_1 with the smallest rating value may be detected by the conventional method as a *top-N* item, while the high-order quantitative preference can capture the high preference of other items well w.r.t. the item i_i.

On the other hand, the high-order quantitative preference can help to better estimate the similarity between users. For instance, suppose that three users u_1, u_2 and u_3 rate items $\{i_1, i_2, i_3, i_4, ...\}$ with the skewed ratings $\{4, 5, 5, 5, ...\}$, $\{1, 2, 2, 2, ...\}$ and $\{5, 4, 4, 4, ...\}$ respectively. From the standpoint of high-order preference, u_1 and u_2 have similar relative preferences, and u_1 and u_3 have totally opposite (relative) preferences. Conventional methods have difficulty in distinguishing the high-order preference of users. In a typical neighborhood-based method, using cosine-based similarity [16] or Pearson correlation similarity [101], the pair of users u_1 and u_2 and the pair of users u_1 and u_3 have almost the same similarity, about 0.98.

[4] For detailed information about specific methods, please refer to the Related Work section.

With the aim of capturing this high-order quantitative preference to solve the recommendation problem for highly skewed datasets, we propose a new CF framework. This framework contains a novel optimization criterion and an effective two-step learning algorithm, which can be applied to learn mainstream neighborhood- or model-based methods. The criterion captures the pointwise rating information and personalized high-order preference simultaneously. The learning algorithm ensures that the framework is general enough to encompass existing CF models and can be readily integrated into current infrastructure in real applications. By utilizing high-order information directly extracted from the rating data, this framework saves the time cost of estimating supplementary parameter vectors and avoids using external information which is not always available.

Unlike the pairwise preference in [94], where user demographic information and item content features are employed to capture the user's preference, our high-order preference is derived directly from available ratings. In contrast to previous work on pairwise ranking [79] and ranking-based CF [99], where the training datasets are purely boolean values and an indicator function is used to characterize the error of (wrong) pairwise ranking prediction, we use the (continuous) difference between the given and predicted rating pairs to measure the preference prediction error.

In designing the learning algorithm, we avoid the high time complexity of using full gradient descent [82], alternating-least-squares (ALS) [54, 133] or an evolutionary approach to optimize the multi-objective criterion [114]. By employing stochastic gradient descent, we make use of the anti-symmetry and transitivity properties of pairwise preference to develop an effective learning algorithm.

In summary, this article makes the following contributions.

1. We present a new general rating and high-order preference framework, RP, for highly skewed rating datasets. This framework makes it possible to create novel and efficient models to capture users' high-order preference among items.

2. To capture characteristics of rating and personalized pairwise preference, we propose a generic optimization criterion OptRP, which is constructed by minimizing the regularized squared error function. And we propose a generic and

effective learning method LEARNRP based on stochastic gradient descent to optimize the criterion, utilizing the property of pairwise preference.

3. We demonstrate that two state-of-the-art CF models can be efficiently learned with RP, and generate two new, more effective models. Empirical results of recommendation tasks on several real-world rating datasets demonstrate the effectiveness of the learned models on highly skewed rating datasets.

The rest of this paper is organized as follows: In Section 2, we briefly review some related work. Section 3 introduces the concepts of rating and personalized high-order preference, and describes our proposed optimization criterion OptRP and learning algorithm LearnRP in the RP framework. Two typical CF models are learned under RP in Section 4. Experiments on real-world rating datasets are discussed in Section 5. Finally, we present our conclusions and some future directions in Section 6.

3.1 Related Work

3.1.1 Approaches for Recommender Systems

Three primary ways of building recommender systems [4] are through collaborative filtering (CF), content-based filtering and a hybrid approach. The CF approach collects and merges user preference information, and generates predictions for an individual user based on similarity measurements of users and items [30, 107, 53, 104]. Related work on CF is presented extensively in the next subsection. Content-based filtering [36, 90, 95, 80] is another common approach to recommender systems, which generates recommendations by utilizing content profiles of items and user profiles that describe the types of item the users like. In other words, this approach tries to recommend items which are similar to those that a user liked in the past. An early study [36] presents a content-based information filtering system, matching user interests to text documents using two matching methods and two types of user profiles. The LIBRA system is a book recommender that uses a Bayesian learning algorithm and extracts information on books for text categorization [90]. Other authors [95, 80] survey the field of content-based recommendation, including a method for representing items and user profiles, and a method for comparing items with the user to determine which to recommend. Content-based filtering is the best approach when content information for users and items is easy to obtain [90]. However, it suffers from limitations in such areas as item representation, serendipitous recommendation and quality assessment of filtering items [109, 20].

The hybrid approach to recommender systems [19] typically combines collaborative and content-based filtering, and can be more effective in some cases. The recommender system implemented on Netflix is an example of this approach. The hybrid approach can be implemented in several ways [4]: for example, by adding content-based characteristics to a collaborative-based method (or vice versa) [12]; or by combining predictions obtained separately using a content-based method and a CF method [88]; or by model unification [97, 14]. Other hybrid methods include Fab, which makes use of profile information to determine similar users for CF [12]; combination of CF and content-based approaches using prediction strengths [88]; probabilistic mixture models [97]; and a kernel-based method which allows generalization across the user and item dimensions simultaneously [14]. Another study [19] surveys the area of possible hybrid recommender systems and examines different types of combinations.

3.1.2 Classic Work on CF

. CF methods are based on the opinions of users rather than the content of items. They are less sensitive to the limitations of content-based filtering mentioned above. Two primary types of CF approaches are widely studied: the neighborhood-based approach and the model-based approach. The neighborhood approach enjoys considerable popularity, due to its simplicity and the explainability of the results [30, 107, 77, 16, 125, 42]. It predicts ratings based on a matrix of similarity values between items [30, 107, 77] or, alternatively, between users [16, 42] or between items and users combined [125, 126]. For instance, the item-based method used in [107] models the preference of a user-item pair based on ratings of similar items assigned by the same user. Methods taking this approach often use cosine-based similarity [16] or Pearson correlation similarity [101] as the similarity measurement.

On the other hand, the model-based approach utilizes the given ratings to train a model, and ratings are then predicted via this model rather than by directly manipulating the original rating data. Many methods adopting this approach have been developed: e.g., the latent semantic model [53], restricted Boltzmann machines [105], and the graph-theoretic model [5]. Matrix factorization methods [104, 102, 65, 66] have been generating much interest and progress recently, due to their attractive accuracy and scalability. This approach transforms both items and users to a joint latent semantic space, and the ratings are estimated by the inner products of a user and an item in that space. For instance, Probabilistic Matrix Factorization (**PMF**) [104] and MF [116] are simple and effective matrix factorization models.

Work to enhance the quality of rating prediction in CF falls into two categories. The first focuses on model improvement and integration, which introduce new parameter vectors into the model for estimation. For instance, matrix-factorization-based models such as SVD [102], SVD++ [65] and timeSVD++ [66] are enhanced by adding user and item biases, implicit user feedback and a time factor, in constructing the models based on MF [116]. The Netflix Prize competition demonstrated the success of this type of improvement. As an example of model integration, mixed-membership matrix factorization (M^{3}F) [83] combines a discrete mixed-membership model with PMF, where the additional topic distribution parameters need to be estimated.

Work in the second category utilizes external data sources concerning the users or the items to yield more accurate predictions. For instance, [120, 121] present a method that incorporates externally specified aggregate rating information into certain types of recommender systems, including model-based and item-based collaborative filtering and hierarchical linear regression (HLM) models. The study in [87] proposes a general model with the incorporation of side information to enhance response prediction quality. RMGM [73] adopts a transfer learning technique and uses the training data of related tasks to advance prediction performance. Work reported in [58, 82, 82, 131] incorporates external information from trust relationship networks [42] or friendship networks among users. Document-centered approaches [111] have been presented for efficient and effective tag recommendation, combining social network information. Another study [118] addresses the music recommendation problem in music social communities, and focuses on combining various types of social media information and music acoustic signals. Recently, the Yahoo music recommendation competition KDD Cup 2011[5], which contains information about songs, such as tracks, albums, artists and genres, has encouraged the development of effective methods by incorporating side information on items.

3.1.3 Related Work on *Top-N* Recommendation.

Top-N recommendation is another fundamental task of recommenders, aimed at finding a set of items that are most appealing to the specific user. The work reported in [61, 30] first presents and evaluates a class of item-based *top-N* recommendations, using item-to-item or item-to-itemset similarities to generate the top recommendations. The authors show that the conditional probability-based item similarity scheme and higher-order item-based models lead to recommender systems that provide reasonably accurate recommendations. A study on novelty and diversity in *top-N* recommendation [56] formulates the trade-off between diversity and matching quality as a binary optimization problem and introduces an evaluation methodology that allows the performance of different methods to be analyzed from the perspective of their ability to recommend novel but relevant items.

Existing work on enhancing the performance of *top-N* recommendation also falls into two groups: studies that utilize the properties of rating values (e.g., rating variance) and those that explore external data resources. Some studies [86, 68] improve on *top-N* recommendation by utilizing more flexible test rating values (not only discrete values) with rating variance [68] or belief distribution [86]. But these studies still focus on the properties of the rating value itself. Another [63] proposes an error-based algorithm that builds a user-item error matrix employing explicit user feedback, where the error between predicted and given ratings also focuses on the characteristics of a single rating value. More recently [58, 133], social trust [42] has been exploited

[5] http://kddcup.yahoo.com/

to improve the quality of *top-N* recommendation. The work reported in [26] evaluates several state-of-the art recommender algorithms and offers two new variants of collaborative filtering algorithms to address the *top-N* recommendation task. The study indicates that improvements in $RMSE$ often do not translate into improved accuracy.

3.2 Rating and Preference (RP) Framework

In this section, a generic RP framework is proposed, which can be applied to learn CF models in the neighborhood-based and model-based categories. We first derive the quantitative pairwise preference, then propose the RP framework, composed of a generic optimization criterion OptRP and a two-step learning algorithm LearnRP.

3.2.1 Rating and Quantitative Pairwise Preference

Let us assume a set $\mathcal{U}$ of n users and a set $\mathcal{I}$ of m items in a typical CF scenario. Each user u is associated with a set $\mathcal{I}_u$, which contains all the items the user has rated. The dataset containing all users and all rated items is denoted by $\mathcal{D}_t \subset \mathcal{U} \times \mathcal{I}$. All observed ratings r_{ui} on the dataset $\mathcal{D}_t$ are denoted by the rating dataset $\mathcal{R}_t$.

$$\mathcal{R}_t := \{r_{ui} | (u,i) \in \mathcal{D}_t\}$$

Now let us investigate the concept of high-order preference developed here. By high-order preference, we mean a user's preference of one item over another. This is personalized information and is therefore captured by pairwise comparison of user ratings. Compared with pairwise preference of items, this personalized pairwise preference is able to capture the information in a user-specific item order reflecting his/her preference, and consequently will enable more meaningful personalized *top-N* recommendation. Below, we derive a scenario for quantitative pairwise preference. The dataset with personalized pairwise preference (u,i,j) is denoted by $\mathcal{D}_n \subset \mathcal{U} \times \mathcal{I} \times \mathcal{I}$.

$$\mathcal{D}_n := \{(u,i,j) | \{(u,i),(u,j)\} \subseteq \mathcal{D}_t\}$$

The difference between a pair of ratings is used to define a quantitative pairwise preference $r_{uij} = r_{ui} - r_{uj}$. The semantic meaning of r_{uij} is the degree to which user u prefers item i to item j, where a high value indicates a stronger preference for i over j. The value can be negative, indicating that user u prefers j to i. The set of all the derived r_{uij} on the dataset $\mathcal{D}_n$ is defined as the personalized pairwise preference set $\mathcal{R}_n$.

$$\mathcal{R}_n := \{r_{uij} | (u, i, j) \in \mathcal{D}_n\}$$

Note that $\mathcal{R}_n$ satisfies anti-symmetry and transitivity, $r_{uij} = -r_{uji}$ and $r_{uij}+r_{ujk} = r_{uik}$. These properties will be utilized to propose an effective learning algorithm in Section 3.2.3.

Now, we utilize the quantitative pairwise preference to define a similarity measure between a user's preference for one item and that for another item. Given a user u and items i and j, r_{uij} measures the difference between user u's preferences for items i and j. Adopting a commonly used procedure, we apply a Gaussian kernel to transform this preference difference into a similarity. This preference similarity can be defined as $s_{uij} = \exp(-r_{uij}^2)$. A small value of s_{uij}, i.e., a value close to zero, indicates that the preferences of user u w.r.t. items i and j are very different, while a large value, i.e., close to one, indicates that the user has very similar preferences towards these two items. The set of all the s_{uij} on dataset $\mathcal{D}_n$ is defined as the preference similarity set $\mathcal{R}_s$.

$$\mathcal{R}_s := \{s_{uij} | (u, i, j) \in \mathcal{D}_n\}$$

3.2.2 RP Optimization Criterion (OptRP)

In this subsection, we first define the rating prediction error and the quantitative preference prediction error, and then propose a regularized squared error function, named the optimization criterion OptRP. This proposed criterion aims to capture, simultaneously, the pointwise rating information and the high-order preference information represented by pairwise preference and pairwise preference similarity.

Here, we define the rating prediction and quantitative preference prediction errors as $x_{ui} = r_{ui} - \hat{r}_{ui}(\Theta)$ and $x_{uij} = r_{uij} - \hat{r}_{uij}(\Theta)$ respectively, where Θ represents the parameter vector of a CF model, and $\hat{r}_{ui}(\Theta)$ and $\hat{r}_{uij}(\Theta)$ are arbitrary real-valued functions of this model. For notational convenience, we drop the argument Θ in the following discussion: e.g., we write $\hat{r}_{ui}$ in place of $\hat{r}_{ui}(\Theta)$. Note that the preference prediction error x_{uij} can be readily combined with the rating prediction error.

$$\begin{aligned} x_{uij} &= (r_{ui} - r_{uj}) - (\hat{r}_{ui} - \hat{r}_{uj}) \\ &= (r_{ui} - \hat{r}_{ui}) - (r_{uj} - \hat{r}_{uj}) \\ &= x_{ui} - x_{uj} \end{aligned}$$

We define the optimization criterion OptRP, which adopts the rating prediction and quantitative preference prediction errors and uses preference similarity as an additional regularization. To learn the model parameters, we just need to minimize OptRP as follows:

$$\begin{aligned} &\sum\nolimits_{\mathcal{R}_t} x_{ui}^2 + \sum\nolimits_{\mathcal{R}_n} \alpha x_{uij}^2 + \lambda_s \sum\nolimits_{\mathcal{R}_n} \alpha s_{uij} \hat{d}_{ij} + \lambda_\Theta \left\|\Theta\right\|^2 \\ = &\sum\nolimits_{\mathcal{R}_t} x_{ui}^2 + \sum\nolimits_{\mathcal{R}_n} \alpha(x_{uij}^2 + \lambda_s s_{uij} \hat{d}_{ij}) + \lambda_\Theta \left\|\Theta\right\|^2 \end{aligned} \tag{3.1}$$

where x_{ui}^2 and x_{uij}^2 denote the errors of an individual rating and pairwise preference, and $\hat{d}_{ij}$ denotes the item difference, which is an arbitrary real-valued function of a CF model that can be defined to measure the difference between items i and j. Note that the error of x_{ui}^2 (or x_{uij}^2) is low for a predicted rating (or pairwise preference) close to the given rating (or pairwise preference), and the error is high for a predicted value far from the given value. In the preference similarity regularization term $\lambda_s \sum_{\mathcal{R}_n} \alpha s_{uij} \hat{d}_{ij}$, a small s_{uij} indicates that the item difference $\hat{d}_{ij}$ is prone to be larger, while a large s_{uij} shows that $\hat{d}_{ij}$ is likely to be smaller. The semantic meaning of this regularization is that a user is likely to have high preference similarity on two similar items and low preference similarity on two different items.

The reasons for using preference similarity as a regularization term to impose constraints on the item difference are twofold. The preference prediction error characterizes the specific preference restriction on each user-item-item pair, and may tend to enlarge the difference between two items if not regularized. This regularization term can be implemented to limit the item difference. In addition, this regularization implicitly captures the difference between items which are not rated together. More specifically, if user u has rated items i and j and user v has rated items j and k (suppose i and k are never rated together by an individual user), we indirectly obtain the item difference of i and k when we minimize the regularization of $s_{uij}\hat{d}_{ij}$ and $s_{vjk}\hat{d}_{jk}$. The propagation of item difference will reach a harmonic status when the values of the criterion converge in the learning phase.

In the OptRP criterion, λ_Θ work as regularization factors for the model parameters to avoid overfitting a model, and λ_s is the regularization factor of the additional preference similarity regularization. The constraints of pairwise preference error and similarity regularization can be viewed as two parts of the preference metric. As suggested in [106], for non-uniform sets $\mathcal{D}_t$ and $\mathcal{D}_n$, we use λ_t and λ_n as the regularization factors λ_Θ for the rating and preference metrics. Therefore, OptRP is a criterion resulting from linearly combining measurements of rating and preference, while the factor α can be treated as the weighting coefficient between two kinds of metrics.

3.2.3 RP Learning Algorithm (LearnRP)

OptRP in Eq. 3.1 is a multi-objective criterion [114] which captures characteristics of pointwise rating, quantitative pairwise preference and pairwise preference similarity. To solve this optimization problem, we could use the full gradient descent [82], the alternating-least-squares (ALS) [54, 133] or the evolutionary approach [114] for this multi-objective criterion. However, these methods are time-consuming. In this work, we employ stochastic gradient descent and make use of properties of pairwise preference to develop an effective learning algorithm.

We propose an efficient learning algorithm LearnRP (Algorithm 3) to optimize the OptRP criterion. Since the pairwise preference error and the regularization of pairwise preference similarity are generated from the same $\mathcal{R}_n$, for effective learning, we implement the constraints of preference similarity as an additional regularization of the preference metric. We adopt different learning rates γ_t and $\gamma_n = \alpha \cdot \gamma_t$ to adjust the weighting of these two metrics in the learning algorithm. In addition, the practical implementation makes use of the anti-symmetry and transitivity properties of pairwise preference $\mathcal{R}_n$ to further reduce the time complexity. This learning algorithm can be used to generate new models which capture pointwise rating and high-order preference information simultaneously, without requiring significant change to the original models and without imposing much time cost.

Algorithm 3 Learning RP (LearnRP)

```
Initialize $\Theta$
repeat
    Draw $(u, i)$ uniformly from $\mathcal{R}_t$
    Draw associated $(u, i, j)$ uniformly from $\mathcal{R}_n$
    $\Theta \leftarrow \Theta + \gamma_n \left( x_{uij} \frac{\partial \hat{r}_{uij}}{\partial \Theta} + \lambda_s e^{-r_{uij}^2} \frac{\partial \hat{d}_{ij}}{\partial \Theta} + \lambda_n \Theta \right)$
    $\Theta \leftarrow \Theta + \gamma_t \left( x_{ui} \frac{\partial \hat{r}_{ui}}{\partial \Theta} + \lambda_t \Theta \right)$
until convergence
return $\hat{\Theta}$
```

To summarize, Fig. 3.2 provides a graphical representation of the RP framework. On the left side, the top matrix indicates the known user-item ratings given by the training dataset and the unknown ratings, marked as '?', which need to be predicted by the learned model. The bottom left tensor shows the transformed personalized pairwise preference information. On the right side are the predicted rating matrix $\hat{R}_t$ and the corresponding converted pairwise preference tensor $\hat{R}_n$. The objective metrics are shown in the middle of this graph, where M_c denotes the proposed multi-objective criterion OptRP. It is composed of the rating-based metric M_t and the preference-based metric M_n. We can employ the LearnRP algorithm to optimize a CF model w.r.t. the criterion OptRP.

3.3 Learning models with RP

The proposed RP is a generic framework. It can be utilized to capture pointwise rating and high-order preference information and generate new, effective CF models. In this section, under the RP framework, we describe how we can learn two mainstream

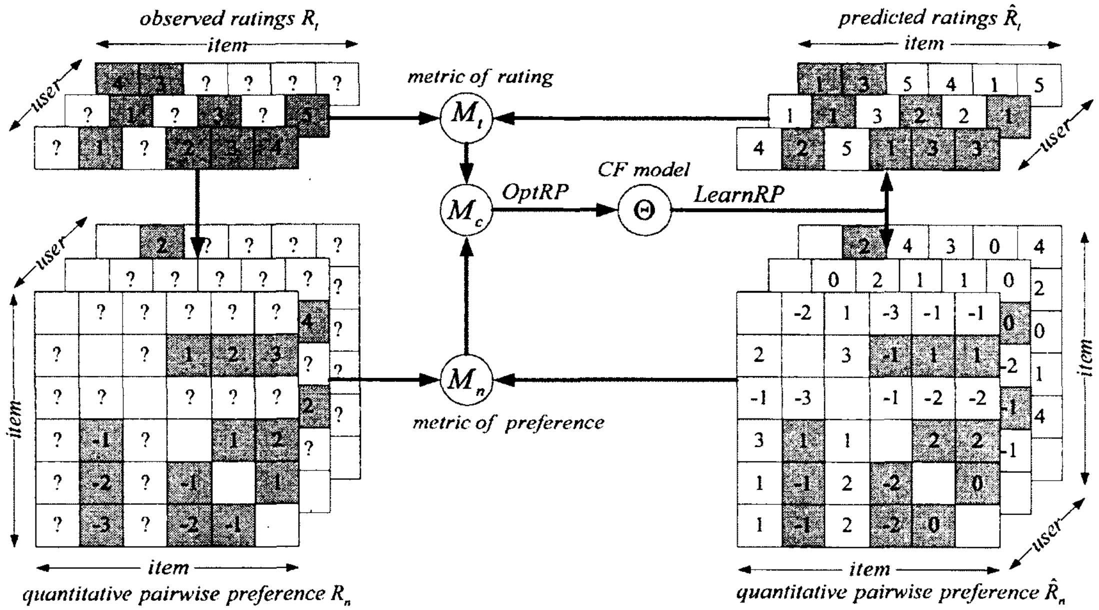


Figure 3.2: Graphical representation of the RP framework. On the left side, the shaded regions are the observed rating set R_t and the personalized pairwise preference R_n. The transformation rule from R_t to R_n is indicated implicitly. The right side shows the predicted datasets. The middle part of this figure represents the rating metric M_t of OptRP, which trains the model from R_t, and the personalized pairwise preference metric M_n, which trains from R_n. These two metrics compose our new OptRP criterion M_c. LearnRP can be used to train CF models w.r.t. OptRP.

models belonging to two primary CF categories, and generate two new, more effective models.

3.3.1 Matrix Factorization Approach

In this subsection, we describe how a mainstream matrix factorization model, MF [116], can be trained under the RP framework. The new learned model is denoted by MF-RP. In a typical MF, the f-dimension factor vectors $p_u \in \mathbb{R}^f$ and $q_i \in \mathbb{R}^f$ describe the latent characteristics of user u and item i, and the predicted rating is calculated as $\hat{r}_{ui} = q_i^T p_u$. To learn MF with RP, we shall calculate the gradients of the rating and preference metrics. First of all, we compute the gradient of predicted rating $\hat{r}_{ui}$ w.r.t. the parameters of the rating metric.

$$\frac{\partial \hat{r}_{ui}}{\partial \theta} = \begin{cases} q_i, & when\ \theta = p_u \\ p_u, & when\ \theta = q_i \end{cases}$$

According to the updating function in Step 6 of Algorithm 3, the detailed updating formulas of all parameters in the rating metric can be expressed as

$$\begin{aligned} q_i &\leftarrow q_i + \gamma_t \left(x_{ui} p_u^T + \lambda_t q_i \right) \\ p_u &\leftarrow p_u + \gamma_t \left(x_{ui} q_i^T + \lambda_t p_u \right) \end{aligned}$$

Then we consider the learning rule in the preference metric. Using the formula for the predicted rating formula $\hat{r}_{ui}$, a quantitative pairwise preference $\hat{r}_{uij}$ can be written as $\hat{r}_{uij} = \hat{r}_{ui} - \hat{r}_{uj} = (q_i - q_j)^T p_u$. Similarly, based on this formula, the gradient of predicted pairwise preference $\hat{r}_{uij}$ w.r.t. the parameters in the preference metric is given by

$$\frac{\partial \hat{r}_{uij}}{\partial \theta} = \begin{cases} p_u, & when\ \theta = q_i \\ -p_u, & when\ \theta = q_j \\ q_i - q_j, & when\ \theta = p_u \end{cases}$$

In the matrix factorization models, we can define the item difference function in the preference metric as $\hat{d}_{ij} = ||q_i - q_j||_{Fro}^2$, where $||\cdot||_{Fro}$ denotes the Frobenius norm. Based on this formula, the gradient of predicted item difference $\hat{d}_{ij}$ w.r.t. the parameters in the preference metric is written as

$$\frac{\partial \hat{d}_{ij}}{\partial \theta} = \begin{cases} q_i - q_j, & when\ \theta = q_i \\ q_j - q_i, & when\ \theta = q_j \\ 0, & when\ \theta = p_u \end{cases}$$

Finally, according to the updating formula in Step 5 of Algorithm 3, the detailed expressions for updating the parameters in the preference metric can be obtained as follows:

$$\begin{aligned}
q_i &\leftarrow q_i + \gamma_n(x_{uij}p_u^T + \lambda_n q_i + \lambda_s e^{-r_{uij}^2}(q_i - q_j)) \\
q_j &\leftarrow q_j + \gamma_n(-x_{uij}p_u^T + \lambda_n q_j + \lambda_s e^{-r_{uij}^2}(q_j - q_i)) \\
p_u &\leftarrow p_u + \gamma_n(x_{uij}(q_i - q_j)^T + \lambda_n p_u)
\end{aligned}$$

3.3.2 Neighborhood Approach

In the work described below, we choose the item-based model [107] to be learned under the RP framework, as it usually provides better prediction, but the user-based model works analogously. There are a variety of different ways [16, 101] to calculate the similarity between items, and we use the cosine-based similarity [16]. This item-based model with the cosine similarity measurement learned with RP is denoted as cosinKNN-RP. The similarity matrix S measures the similarities between the items in the set $\mathcal{I}$, where s_{ij} denotes the similarity of item i and item j. The matrix S is symmetric and all values in its diagonal are one. In concrete terms, the cosine-based similarity s_{ij} represents the cosine of the angle between two item vectors $\vec{i}$ and $\vec{j}$ in an n-dimension user-space. It is calculated by

$$s_{ij} = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \cdot \vec{j}}{||\vec{i}||_2 * ||\vec{j}||_2}$$

The idea of the item-based model is to find a set $\mathcal{I}_u^k$ containing the k most similar items that the target user has rated, after which the prediction is generated by computing a weighted average of the user's ratings on these similar items.

$$\hat{r}_{ui} = \frac{1}{\sum_{m \in \mathcal{I}_u^k} |s_{im}|} \sum_{m \in \mathcal{I}_u^k} s_{im} \cdot r_{um} \tag{3.2}$$

Again, for learning the model under the RP framework, the gradient of predicted rating $\hat{r}_{ui}$ w.r.t. the parameters of the rating metric can be calculated by

$$\frac{\partial \hat{r}_{ui}}{\partial \theta} = \frac{r_{um} - \hat{r}_{ui}}{\sum_{m \in \mathcal{I}_u^k} |s_{im}|}, \quad when\ \theta \in \{s_{im}, s_{mi}\}.$$

This yields the update formula for s_{im}; s_{mi} can also be updated by this formula.

$$s_{im} \leftarrow s_{im} + \gamma_t(x_{ui} \cdot \frac{r_{uk} - \hat{r}_{ui}}{\sum_{m \in \mathcal{I}_u^k} |s_{im}|} - \lambda_t s_{im})$$

The predicted quantitative pairwise preference $\hat{r}_{uij}$ can be calculated from Eq. 3.2. The gradient of $\hat{r}_{uij}$ w.r.t. the parameters of the preference metric can then be expressed as

$$\frac{\partial r_{uij}}{\partial \theta} = \begin{cases} \frac{r_{um} - \hat{r}_{ui}}{\sum_{m \in \mathcal{I}_u^k} |s_{im}|}, & when\ \theta \in \{s_{im}, s_{mi}\}\,, \\ -\frac{r_{um} - \hat{r}_{uj}}{\sum_{m \in \mathcal{I}_u^k} |s_{jm}|}, & when\ \theta \in \{s_{jm}, s_{mj}\}\,. \end{cases}$$

In the neighborhood-based models, we can define the item difference function in the preference metric as $\hat{d}_{ij} = \|s_i - s_j\|_{Fro}^{-2}$. Based on this formula, the gradient of predicted item difference $\hat{d}_{ij}$ w.r.t. the parameters in the preference metric is written as

$$\frac{\partial \hat{d}_{ij}}{\partial \theta} = \begin{cases} -2\,\|s_i - s_j\|_{Fro}^{-3}, & when\ \theta \in \{s_{im}, s_{mi}\} \\ 2\,\|s_i - s_j\|_{Fro}^{-3}, & when\ \theta \in \{s_{jm}, s_{mj}\} \end{cases}$$

Analogously, the specific expressions for updating the parameters s_{im} and s_{jm} in the preference metric can be deduced; s_{mi} and s_{mj} can also be updated by the formulas for s_{im} and s_{jm} respectively.

$$s_{im} \leftarrow s_{im} + \gamma_n \left(x_{uij} \frac{r_{um} - \hat{r}_{ui}}{\sum_{m \in I_u^k} |s_{im}|} + \lambda_n s_{im} - 2\lambda_s e^{-r_{uij}^2} \|s_i - s_j\|_{Fro}^{-3} \right)$$

$$s_{jm} \leftarrow s_{jm} + \gamma_n \left(-x_{uij} \frac{r_{um} - \hat{r}_{uj}}{\sum_{m \in I_u^k} |s_{jm}|} + \lambda_n s_{jm} + 2\lambda_s e^{-r_{uij}^2} \|s_i - s_j\|_{Fro}^{-3} \right)$$

3.4 Experiments and analysis

In this section, we conduct effectiveness and efficiency tests to analyze the performance of the models learned under the proposed RP framework. To test effectiveness, we evaluate rating prediction and *top-N* recommendation performance on three public rating datasets. For the efficiency test, we compare the time consumption of the learned models and the original models on these datasets. Moreover, we examine how predictive performance is affected by the coefficient α, corresponding to the relative weighting of rating and preference metrics.

3.4.1 Experiment Design

Datasets. For our experiments, we adopt three public rating datasets, the Epinions, Amazon and Ciao datasets. The first concerns articles and the latter two are associated with the product domain.

In the extended Epinions dataset[6] provided by the authors of [85], each data item reflects how highly an individual user rates a certain textual review written by another user. There are $132,000$ users who have provided about $1,560,144$ articles with 13 million ratings in the range $[1,5]$ (from "not helpful" to "very helpful"). In this dataset, some ratings with value six are treated as five, as the author suggested.

From the Stanford Network Analysis Project (SNAP)[7], we downloaded the Amazon meta-information for $437,554$ products. We cleaned the meta-information and generated the user-product rating dataset, where the rating values indicate the preferences of users w.r.t. a certain number of products on Amazon. There are $1,555,152$ users who have provided about $437,554$ products with $6,409,755$ ratings in the range $[1,5]$.

The Ciao dataset[8] crawled in May 2011 describes the preferences of $7,375$ users w.r.t. $106,797$ products, where about $284,086$ million ratings are given in the range $[1,5]$ with a one-star increment. Compared with the highly-skewed Epinions and Amazon datasets, the skewness measurement of this rating dataset is much lower. We chose these three datasets with different skewness to examine the performance improvement w.r.t. the skewness measurement.

Evaluation methodology. To examine the effectiveness of the RP framework on

[6] http://www.trustlet.org/wiki/Extended_Epinions_dataset

[7] http://snap.stanford.edu/data/amazon-meta.html

[8] http://www.public.asu.edu/ jtang20/datasetcode/truststudy.htm

the tasks of rating prediction and *top-N* recommendation simultaneously and analyze the relation between these two tasks, we compare the learned MF-RP and cosinKNN-RP models mentioned above with their natural competitors, i.e., the original MF and cosinKNN models. The experiment is designed as follows. From the experimental datasets, we randomly choose fifteen ratings of users to compose the test datasets, and the remaining ratings are treated as the training datasets. The tests are repeated 10 times and the average performance results are calculated. For all models, the exact learning rates and hyperparameters are determined by cross-validation on the training datasets.

To measure rating prediction performance, we adopt two different evaluation metrics, $RMSE$ and $NDCG$, which belong to predictive accuracy measurement and predictive rank measurement, respectively [51]. The predictive accuracy performance is measured by the classic and widely-adopted rating-based criterion $RMSE$ [65, 82, 102]. Values close to zero show better performance.

$$RMSE = \sqrt{\frac{\sum_{(u,i)\in R_t} (r_{ui} - \hat{r}_{ui})^2}{|R_t|}}$$

Moreover, we use a rank accuracy metric to evaluate the predictive performance from the item ranking standpoint. As mentioned in [51], a rank accuracy metric measures the ability of a recommender to generate an order of items that corresponds to the way the user would have ordered the same items. In this work, we implement the Normalized Discounted Cumulative Gain (**NDCG**) [79, 130] to examine the ranking quality of the rating predictions.

$$NDCG@N = \frac{1}{|\mathcal{U}|} \sum_{u=1}^{|\mathcal{U}|} \frac{1}{Z_u} \sum_{n=1}^{N} \frac{2^{r_u^n} - 1}{\log_2(n+1)}$$

where $|\mathcal{U}|$ indicates the number of users associated with more than fifteen ratings, Z_u is a normalization factor for user u, and r_u^n is the ground truth rating value of the item at the position n predicted by the algorithms. We calculate the results (from $NDCG@1$ to $NDCG@10$) to represent the rank accuracy of prediction on the first

10 items. Note that a larger *NDCG* value indicates a better result.

For evaluation of *top-N* recommendation, we use the overall *Recall* value which is calculated by averaging over all test users. From the fifteen ratings of each test user, we choose the five items with the highest rating values as the objective of the top recommendation, with regards to the last five items with smallest ratings of this user. In our experiment, we avoid the assumption, made in [26][63], that the item with the highest rating is more interesting to the user than other unrated items. This assumption is not always true. Another study [84] reveals that missing ratings cannot be assumed to be small and unrated items can also be the most interesting items for a specific user. Similar to previous evaluation work [61, 58] on top-N recommendation, the performance is measured by the *Recall* criterion [65, 82, 102]. The *Recall* measurement is expressed as follows:

$$Recall@N = \frac{1}{|\mathcal{U}|}\sum_{u=1}^{|\mathcal{U}|}\frac{\#hits(u,N)}{5\,|\mathcal{U}|}$$

where $\#hits(u, N)$ indicates how many of the u's *top*-5 items of the test dataset appear in the predicted list within the length $N(5 \leq N \leq 9)$. When $N = 10$, all *Recall* results are 1; these results are not included in our result list. Larger *Recall* values indicate better *top-N* recommendation.

Reproducibility. To ensure experimental reproducibility, all of the datasets chosen are public datasets. Our MF-RP and cosinKNN-RP implementations are available upon request by email. Here, we list the hyperparameter settings of the MF-RP model on these datasets. In all datasets, the latent vectors p_u and q_i are initialized with random values drawn from a normal distribution $N(0, 0.1)$, the coefficient α=0.01 and the regularization factor λ_t=λ_n. The specific hyperparameter settings are as follows: in the Epinions dataset, the learning rate γ_t=0.005, and regularization factors λ_t=0.02 and λ_s=64; in the Amazon dataset, γ_t=0.01, λ_t=0.05 and λ_s=4; in the Ciao dataset, γ_t=0.1, λ_t=0.1 and λ_s=512. In the neighborhood models, we use all items as the set of neighbors [65], which means all the item-item similarities are examined.

3.4.2 Convergence Rate

Improved convergence rates were observed with the MF-RP model compared to the original MF model. Fig. 3.3 shows the convergence comparison for MF-RP and the original MF on the Epinions and Amazon datasets with 10-dimensional factor vectors. As we can see, the MF-RP model converges similarly to the original MF model at the early stage. On these two datasets, however, as of a certain point (around iteration 100 for both datasets), the MF model exhibits the phenomenon of over-training, while the MF-RP model continues to improve toward a minimum RMSE. In all our experiments, in contrast to the original MF model, the MF-RP model never suffers from over-training.

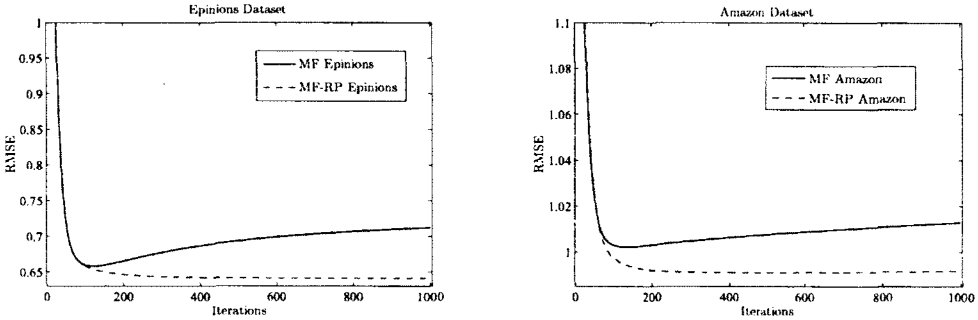


Figure 3.3: Empirical comparison of the convergence of MF and MF-RP on the Epinions and Amazon datasets, with f=10.

3.4.3 Effectiveness Analysis

The prediction accuracies of the matrix factorization and neighborhood models, measured by $RMSE$, are respectively illustrated in Table 3.1 and Table 3.2. These results indicate that by capturing quantitative pairwise preference, the learned MF-RP and cosinKNN-RP models tend to yield better performances than the original MF and cosinKNN models, respectively.

We begin by analyzing the results of the models in the matrix factorization category. The rating prediction results in Table 3.1 show that all of the matrix factorization models benefit from an increased number of factor dimensions f, which can better capture the latent characteristics of the users and items. In these three datasets, the advantage consistently delivered by MF-RP over MF is significant with the greater f, due to the additional preference constraints resulting from the new learning framework. Further evidence of the importance of the preference constraints in our framework is the fact that MF-RP at f=10 for the high-skewed Epinions dataset is already more accurate than the MF model at f=100. In the neighborhood model category, cosinKNN-RP can be viewed as an improvement on the basic cosinKNN model with additional constraints of quantitative pairwise preference. The results on these datasets presented in Table 3.2 show that cosinKNN-RP greatly outperforms the basic cosinKNN, and the performance of learned cosinKNN-RP on the Ciao dataset is close to that of the matrix factorization models.

Table 3.1: Accuracy of rating prediction on the two datasets, measured by *RMSE*, for varying dimensionality f.

	f	10	20	50	100
Epinions	MF	0.6683	0.6678	0.6675	0.6659
	MF-RP	**0.6517**	**0.6514**	**0.6514**	**0.6486**
Amazon	MF	1.0688	1.0213	0.9824	0.9720
	MF-RP	**1.0510**	**1.0041**	**0.9741**	**0.9653**
Ciao	MF	1.8748	1.8767	1.8691	1.7395
	MF-RP	**1.8618**	**1.8626**	**1.8543**	**1.7155**

Table 3.2: Performances of neighborhood models evaluated by *RMSE*.

Data	CosinKNN	CosinKNN-RP
Epinions	1.1130	**0.9597**
Amazon	1.6545	**1.2643**
Ciao	1.9321	**1.8964**

The prediction accuracies of the MF and MF-RP models for different rating values, evaluated by *RMSE*, are listed in Table 3.3. As indicated, the MF-RP model tends to generate more precise predictions on the ratings at the high end of the range, i.e., ratings 4 and 5. Since this portion of the range contains most of the ratings in the whole dataset (about 90%), MF-RP greatly improves on the predictive performance of MF. Moreover, since the items with high ratings are the most appealing items for users, these ratings are more significant than the ratings at the low end of the range. Higher predictive accuracy for ratings at the high end of the range also leads to more precise *Top-N* recommendation.

Table 3.3: Accuracy of rating prediction on Epinions w.r.t. different rating values when the dimensionality $f = 10$ and $f = 100$, evaluated by *RMSE*.

Rating	1	2	3	4	5
Percentage of rating	0.011%	4.422%	4.675%	**13.116%**	**77.776%**
MF f=10	**2.2525**	**1.6076**	**1.0690**	0.6495	0.5185
MF-RP f=10	2.2613	1.6211	1.0745	**0.6283**	**0.4945**
MF f=100	**2.1453**	**1.5531**	**1.0321**	0.6288	0.5356
MF-RP f=100	2.2432	1.6057	1.0675	**0.6212**	**0.4928**

User-specific skewness measurement and observed improvement. Skewness [62] is the third standardized moment measuring the asymmetry of the data around the sample mean. Since the negative or positive values just describe the long tail on the left or the right side, we use the absolute skewness as the asymmetry measurement in this work. The absolute skewness of the rating distribution of a user u, denoted by s_u, is defined as follows:

$$s_u = \frac{|m_3|}{m_2^{3/2}} = \frac{\left|\frac{1}{|\mathcal{I}_u|}\sum_{i\in\mathcal{I}_u}(r_{ui}-\bar{r}_u)^3\right|}{\left(\frac{1}{|\mathcal{I}_u|}\sum_{i\in\mathcal{I}_u}(r_{ui}-\bar{r}_u)^2\right)^{3/2}}$$

where m_e is the eth central moment of the mean and $\bar{r}_u$ is the mean rating of u in the training data. Note that a large value of s_u indicates a more asymmetric and long-tailed distribution. Symmetric distributions, e.g., normal distribution and uniform distribution, have the smallest absolute skewness, which is zero. We utilize the average s_u of all users to represent the skewness measurement of a rating dataset

$$\mathcal{S} = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}} s_u$$

There is a correlation between the skewness measurement and the performance improvement of MF-RP over MF. From our preliminary experiments and the work described here, we have observed that the performance improvement of MF-RP over MF on different datasets can, to some extent, be explained by the skewness measurement. The skewness measurements and the average improvement percentages of MF-RP over the original MF model on all dimensionalities are listed in Table 3.4.[9] The Ciao dataset with the lowest skewness measurement shows the smallest percentage of improvement. The Epinions and Amazon datasets have higher skewness measurements than the Ciao dataset, and the improvements on these datasets are greater than on the Ciao dataset. The Epinions dataset has the highest user-specific skewness measurement, and the enhancement on this dataset is the greatest among these datasets. This evidence shows that to some extent, the skewness measurement can be used to explain the performance improvement of MF-RP over MF, and can also be used as an indicator of how much information for a rating dataset can be revealed by the high-order preference.

Table 3.4: Skewness measurements of three datasets and improvements of MF-Brr over MF.

	Epinions	Amazon	Ciao
Skewness	1.7014	1.3387	0.7214
Improvement	2.5509%	1.2517%	0.9035%

Rank accuracy results. Rank accuracy performance [51] reflects how closely the order of personalized predicted items corresponds to the order of user-specific ground truth ratings. The experimental results of rank accuracy measured by $NDCG$ on three rating datasets are given in Table 3.5. We use a two-tailed significance level [32] of 0.002 as detecting ties (statistically similar results) between the results of the original and learned models. The **bold-faced** value in Table 3.5 indicates better performance achieved by the corresponding original or learned model.

In the matrix factorization model category, generally, the MF-RP model yields better performance than the corresponding MF model on all these datasets. Like the predictive accuracy, the rank accuracy of the matrix factorization models increases

[9] The improvements for neighborhood models are not listed, since these improvements contain two components, one resulting from the adaptive approach and the other from the RP framework.

with the dimensionality, and matrix factorization models outperform neighborhood models in terms of rank accuracy prediction. In contrast to predictive accuracy, where the highest improvement of MF-RP over MF is achieved on the Epinions dataset, on the rank accuracy, the greatest improvement of MF-RP over MF is on the Amazon dataset. In the neighborhood model category, the rank accuracies of these datasets are all significantly improved by adding the preference constraints. Table 3.5 indicates that almost all the results of CosinKNN-RP are significantly better than those yielded by CosinKNN.

***Top-N* recommendation results.** Evaluation of *top-N* recommendation reveals the ability of the recommendation algorithms to provide the most appealing items to a specific user. In this paper, we employ the measurement *Recall* to quantify the performance of the competing models. The results of evaluation on these three datasets are listed in Table 3.6, where we set the significance level of 0.002 and the **bold-face** value indicates the better performance achieved by the corresponding original or learned model.

In the matrix factorization model category, the MF-RP model always performs better than the corresponding MF model. MF-RP shows a greater improvement over MF at f=100 than at f=10. Similar to the rank accuracy, the improvement in *top-N* recommendation on the Epinions dataset is again the greatest among these datasets. In the neighborhood model category, the *top-N* recommendations on the datasets are all significantly improved by adding the preference constraints. Table 3.6 indicates that on the Epinions and Amazon datasets the results of CosinKNN-RP are significantly better than the results yielded by the CosinKNN. The improvement in *top-N* recommendation on the Ciao dataset is not as great as on the other two datasets.

To summarize the effectiveness analysis, capturing the high-order preference, the models learned under the RP framework tend to provide better performance than the original models evaluated by three different kinds of measurements. The improvements in prediction accuracy are related to the skewness measurements of the datasets. On the other hand, while the learned models improve performance on three

Table 3.5: Rank accuracy results of the matrix factorization and neighborhood models on three rating datasets, evaluated by *NDCG*.

		NDCG@1	NDCG@2	NDCG@3	NDCG@4	NDCG@5	NDCG@6	NDCG@7	NDCG@8	NDCG@9	NDCG@10
Epinions	MF f=10	0.9082	0.9127	0.9158	0.9183	0.9202	0.9220	0.9242	0.9265	0.9294	0.9330
	MF-RP f=10	**0.9136**	**0.9174**	**0.9203**	**0.9219**	**0.9236**	**0.9252**	**0.9270**	**0.9290**	**0.9316**	0.9349
	MF f=100	0.9125	0.9167	0.9192	0.9213	0.9231	0.9247	0.9265	0.9287	0.9314	0.9348
	MF-RP f=100	**0.9167**	**0.9200**	**0.9221**	**0.9234**	0.9249	0.9264	0.9280	0.9300	0.9325	0.9358
	CosinKNN	0.8684	0.8709	0.8751	0.8738	0.8746	0.8781	0.8821	0.8852	0.8894	0.8951
	CosinKNN-RP	**0.9006**	**0.8922**	**0.8806**	**0.8810**	**0.8790**	0.8798	**0.8856**	**0.8889**	**0.8945**	**0.9008**
Amazon	MF f=10	0.8648	0.8661	0.8673	0.8691	0.8713	0.8744	0.8786	0.8837	0.8897	0.8966
	MF-RP f=10	**0.8738**	**0.8740**	**0.8748**	**0.8760**	**0.8775**	**0.8800**	**0.8836**	**0.8885**	**0.8941**	**0.9008**
	MF f=100	0.9173	0.9146	0.9114	0.9099	0.9094	0.9100	0.9120	0.9151	0.9191	0.9241
	MF-RP f=100	**0.9234**	**0.9197**	**0.9162**	**0.9139**	**0.9129**	**0.9133**	**0.9151**	**0.9180**	**0.9218**	**0.9265**
	CosinKNN	0.7616	0.7464	0.7467	0.7402	0.7416	0.7498	0.7546	0.7659	0.7748	0.7888
	CosinKNN-RP	**0.7960**	**0.8079**	**0.8155**	**0.8256**	**0.8277**	**0.8349**	**0.8361**	**0.8385**	**0.8446**	**0.8545**
Ciao	MF f=10	0.8129	0.8097	0.8054	0.8010	0.7992	0.7991	0.8010	0.8050	0.8121	0.8209
	MF-RP f=10	**0.8184**	**0.8127**	**0.8077**	**0.8031**	0.8000	0.7997	0.8015	0.8058	0.8130	0.8220
	MF f=100	0.8194	0.8105	0.8050	0.8018	0.7995	0.7991	0.8008	0.8049	0.8122	0.8209
	MF-RP f=100	**0.8275**	**0.8185**	**0.8100**	**0.8053**	**0.8025**	**0.8022**	**0.8035**	**0.8074**	**0.8143**	**0.8236**
	CosinKNN	0.7868	0.7777	0.7744	0.7732	0.7715	0.7675	0.7719	0.7742	0.7831	0.7903
	CosinKNN-RP	**0.8037**	**0.8027**	**0.7952**	**0.7937**	**0.7897**	**0.7902**	**0.7904**	**0.7942**	**0.8024**	**0.8119**

Table 3.6: Effectiveness on *top-N* recommendation on three rating datasets, evaluated by *Recall*.

		MF f=10	MF-RP f=10	MF f=100	MF-RP f=100	CosinKNN	CosinKNN-RP
Epinions	Recall@5	0.6102	**0.6179**	0.5982	**0.6172**	0.6052	**0.6128**
	Recall@6	0.6982	**0.7033**	0.6916	**0.7030**	0.6938	**0.6966**
	Recall@7	0.7847	0.7865	0.7817	**0.7865**	0.7794	**0.7821**
	Recall@8	0.8652	0.8653	0.8650	0.8654	0.8590	0.8608
	Recall@9	0.9384	0.9380	0.9377	0.9379	0.9223	**0.9256**
Amazon	Recall@5	0.6759	**0.6823**	0.7210	**0.7326**	0.6645	**0.6917**
	Recall@6	0.7671	**0.7727**	0.8033	**0.8149**	0.7611	**0.7843**
	Recall@7	0.8405	**0.8458**	0.8679	**0.8771**	0.8428	**0.8602**
	Recall@8	0.9014	**0.9046**	0.9200	**0.9265**	0.9101	**0.9209**
	Recall@9	0.9522	0.9539	0.9621	**0.9653**	0.9626	**0.9676**
Ciao	Recall@5	0.6679	**0.6709**	0.6676	**0.6735**	0.6574	**0.6609**
	Recall@6	0.7622	0.7638	0.7610	**0.7650**	0.7523	0.7538
	Recall@7	0.8407	0.8420	0.8385	**0.8428**	0.8397	0.8412
	Recall@8	0.9041	0.9052	0.9033	0.9050	0.8987	0.9003
	Recall@9	0.9547	**0.9571**	0.9567	0.9567	0.9507	0.9527

tasks simultaneously, the relationship between the three kinds of evaluations is nontrivial. A higher prediction accuracy or rank accuracy may not translate into a better *top-N* recommendation. For instance, on Epinions, the MF-RP models yield the best prediction accuracy evaluated by $RMSE$ and the best rank prediction quantified by $NDCG$ of the three datasets, but their relative performances on *top-N* recommendation are reversed. In addition, the performance improvement in prediction or rank accuracies with increasing dimensionality f may not translate into a better *top-N* recommendation. For instance, on Epinions, the rating prediction accuracy evaluated by $RMSE$ and $NDCG$ increases with dimensionality, while the *top-N* recommendation performance even slightly decreases with dimensionality, from $f = 10$ to $f = 100$.

3.4.4 Impact of Coefficient α

In order to understand the effects of the rating-based and preference-based factors on the results of rating prediction and *top-N* recommendation, we examine the prediction performance with different values of the weighting coefficient α. The coefficient $\alpha = \sigma_n^2\sigma_t^{-2}$, implicitly indicated in Algorithm 3 as $\alpha = \gamma_n\gamma_t^{-1}$, modulates the relative weighting of the rating-based and preference-based metrics. In the special case where the coefficient α is set to 0, a model learned under the RP framework corresponds

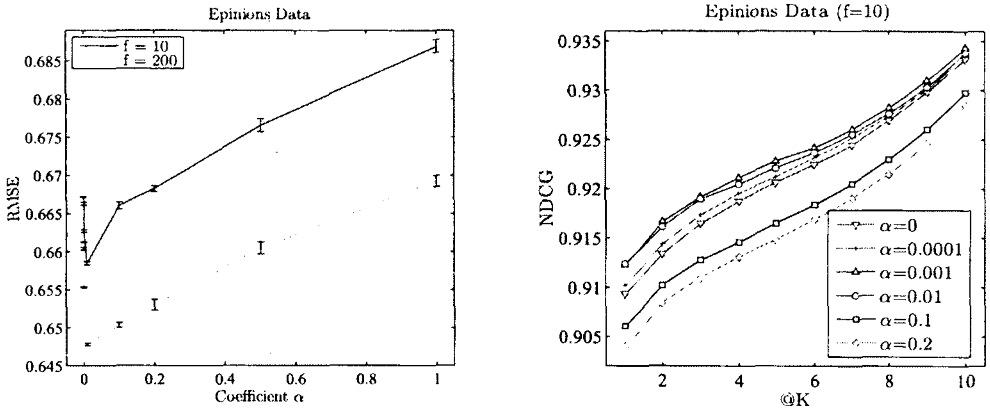


Figure 3.4: Prediction accuracy and rank accuracy of MF-RP on the Epinions dataset, measured by $RMSE$ (α = 0, 0.0001, 0.001, 0.01, 0.1, 0.2, 0.5 and 1) and $NDCG$ respectively, with varying coefficient α.

to the conventional one with only the rating-based metric. Conversely, a large value of α imposes a strong preference-based restriction on the learned model. For our experiment, we vary the coefficient α in the range of $[0, 1]$ to adjust the importance given to the preference metric. The rating prediction and *top-N* recommendation performances on the Epinions dataset with varying coefficient α are shown in Fig. 3.4 and Table 3.7.[10]

For prediction accuracy, the impact of the coefficient α with different dimensionality f on the Epinions dataset is plotted in the left part of Fig. 3.4. The empirical results indicate that α should be set at about 0.01 to yield the best performance on $RMSE$. Generally, when the coefficient α of MF-RP is increasing in the range of $[0, 1]$, the predicted results with different dimensionalities f initially improve in the range $[0, 0.01]$, and then decrease gradually in the range $(0.01, 1]$. The results in the range $[0.0001, 0.1]$ with $f = 10$ and in the range $[0.0001, 0.5]$ with $f = 200$ are better than the results yielded by the model learned solely on the rating-based metric when $\alpha = 0$. Thus, when the dimensionality f increases, the coefficient α can be chosen in a broader ange to make the learned model yield better results than the original

[10] Since the performances of $NDCG$ and *Recall* decrease monotonically when α increases from 0.2 to 1, these results are not included in the illustration.

model. This figure also indicates that the variance in performance is increased with the increasing α. Therefore, in the hyperparameter search, we should set α to a small value (about 0.01) and tune this value in a narrow range to generate the best predictive accuracy.

The results for rank accuracy are illustrated in the right part of Fig. 3.4. When α is set to a small value, i.e., 0.0001, 0.001 or 0.01, the $NDCG$ results are better than the results using the rating-based metric alone (α=0). When α is large, the ranking quality is decreasing and worse than the results achieved by the original model. In contrast to predictive accuracy, where the best results were obtained at $\alpha = 0.01$, the best $NDCG$ results are achieved when $\alpha = 0.001$. These observations implicitly indicate that predictive accuracy and rank accuracy are related but do not totally reflect the same properties of the predictions.

Table 3.7: *Top-N* predictions of MF-RP with varying coefficient α, evaluated by *Recall*.

α	0	0.0001	0.001	0.01	0.1	0.2
Recall@5	0.5830	0.5841	0.5838	0.5882	**0.5884**	0.5846
Recall@6	0.6861	0.6869	0.6867	**0.6891**	0.6885	0.6861
Recall@7	0.7796	**0.7808**	0.7797	0.7807	0.7800	0.7794
Recall@8	0.8640	**0.8644**	**0.8644**	0.8635	0.8621	0.8627
Recall@9	0.9382	**0.9384**	0.9378	0.9372	0.9371	0.9369

For *top-N* recommendation, the performances of MF-RP (f=200) with varying α are shown in Table 3.7. The performances of top-N recommendation at $0.0001 \leq \alpha \leq 0.1$ are better than the performance at α=0 with rating-based metric constraints only. When the coefficient α increases, the model tends to achieve better results at top positions of the prediction list. In contrast to the best predictive accuracy, which was generated at α=0.01, MF-RP yields the best *top-N* recommendation when α=0.0001. This evidence also verifies the non-trivial relationship between these different evaluations.

In the experiment varying the coefficient, we find that the three evaluations attain their best results at different values of α. The coefficient α can be set in the range $[0.0001, 0.1]$, to make the performance quality of the MF-RP measured by all three

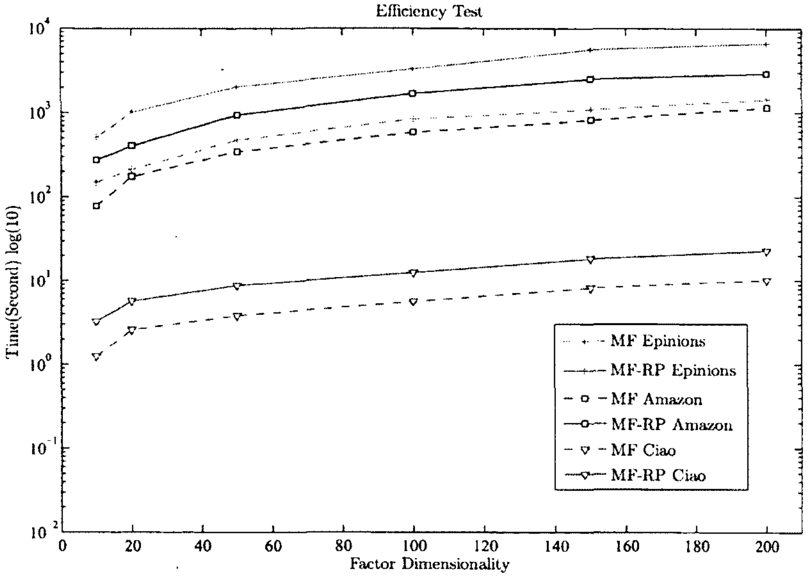


Figure 3.5: Efficiency results of models in the MF group with $f = 10$.

of these evaluations better than the predictive performance yielded by the original model MF learned with the rating-based metric alone. Moreover, to achieve the best performance of a particular evaluation, the coefficient α should be tuned within this range.

3.4.5 Efficiency Analysis

To assess the efficiency of the learned models, we first analyze their time complexity and then calculate the time consumption of the corresponding models on all of the rating datasets. All of the compared models were implemented with C++, and run on a desktop with Intel core i7-2600 processor (clocked at 3.4 GHz) and 8G memory.

First, let us examine the time complexity and the time consumption of the matrix factorization models. MF-RP mainly comprises rating and preference learning steps. In one iteration over the training dataset, the time complexity of the preference constraint learning and that of the rating metric learning are both $O(|\mathcal{R}_t|f)$. Therefore the overall complexity of MF-RP can be expressed as $O(|\mathcal{R}_t|f)$, which is the same as for MF with rating-based metric only. The run times of the matrix factorization models are shown in Fig. 3.5.[11] It can be seen that the time costs of MF and MF-RP are almost linearly increasing with the dimensionality f. Compared with MF, MF-RP needs to update with the supplementary preference metric. The time cost of MF-RP is never more than ten times that of MF.

Now, we will analyze the time complexity of cosinKNN-RP and cosinKNN. The time complexity of cosinKNN-RP is the time complexity of CosinKNN with the time complexity of the loop training step added. Therefore, the time complexity of CosinKNN-RP is $O(|\mathcal{I}|^2|\mathcal{U}| + |\mathcal{R}_t||\mathcal{I}|)$. Since the similarity calculation is more time-consuming than a loop training, the time complexity of CosinKNN-RP can be written as $O(|\mathcal{I}|^2|\mathcal{U}|)$, which is the same as the time complexity of CosinKNN. The run times of the neighborhood models are indicated in Table 3.8. Compared with cosinKNN, the learned cosinKNN-RP models impose less time cost to enhance the effectiveness of rating prediction. In our experiments, the cosinKNN-RP model needs just one or two loops on the training datasets besides the time cost of the cosinKNN model. Compared with the run time of the item similarity computation, the run time for one loop is small.

Table 3.8: Time consumption of CosinKNN and CosinKNN-RP on three datasets. Time (hour)

	CosinKNN	CosinKNN-RP
Epinions	4.7466	5.3922
Amazon	2.0667	2.3167
Ciao	0.8415	0.9142

From the above efficiency tests and complexity analysis, we can see that our learned models do not entail a great increase in time cost compared to the original models. Without imposing much time cost, the RP framework can be widely used to learn models with different scales of time complexity.

[11] To better illustrate the results, in Fig. 3.5, the time consumption axis is presented on the $log(10)$ scale.

3.5 Conclusion

To address the problems conventional models have in dealing with highly-skewed rating datasets, which are frequently encountered, we have proposed a general RP framework with a new criterion OptRP and a corresponding learning algorithm LearnRP to capture high-order preference in this kind of rating datasets. We have shown how RP can be used to generate new, more effective matrix factorization and neighborhood models, which belong to two primary categories of CF. The empirical results on datasets from different domains demonstrate the effectiveness of the learned models, not only for rating prediction but also for *top-N* recommendation. In the future, we would like to examine the effectiveness of non-linear combinations, and plan to explore genetic models for this multi-objective learning framework, where the learning order and coefficients can be tuned implicitly.

Chapter 4

Conclusion and Future Work

This thesis studies two important problems in outlier detection and recommendation systems, which are outlier detection in large-scale categorical datasets and recommendation systems for highly-skewed rating datasets.

For outlier detection in large-scale categorical datasets, we provide a formal definition of an outlier by using the information theory, and propose two effective and efficient algorithms. In fact, to avoid high time consumption, we derive a new outlier factor function and show that computation/updating of the outlier factor is solely determined by the object itself and can be performed efficiently without the need to estimate the joint probability distribution. Experimental results indicate that our proposed algorithms have a linear time complexity with the size of datasets, i.e. the number of objects and dimensions of the datasets, and need only the number of outliers as the input parameter.

For the recommendation task on highly-skewed rating datasets, we first examine the properties of this kind of rating datasets, and then propose a new framework for estimating the rating and quantitative high-order preference. Besides, the transitive associations among the items which are never rated together can be implicitly captured by the constraints of high-order preference similarity in this framework. Experimental results on typical highly-skewed datasets show that new models generated under this framework can generate better performance than the conventional methods not only on rating prediction but also on *Top-N* recommendation.

In the future, based on the research on the outlier detection and recommendation systems, we plan to establish a more effective and robust statistical model which can better capture the reliabilities of user-item ratings.

The rating values are not always objective and reliable. The rating of a specific user on a item may vary greatly when he/her rerates this item [9]. This work indicates that users tend to be inconsistent and introduce a non-negligible amount of natural noise in their ratings that affects the accuracy of the predictions. Based on the obtained re-ratings of some items, [9] provides a strategy to remove a part of natural noise in the pre-processing step of outlier detection. On the other hand, shilling a recommender system for fun or profit is unavoidable in e-commerce websites [69], and the quality of rating in this situation becomes more questionable. As mentioned in [69], "unscrupulous producers in the never-ending quest for market penetration may find it profitable to shill recommender systems by lying to the systems in order to have their products recommended more often than those of their competitors".

To deal with unavoidable inconsistent ratings and ubiquitous shilling attacks, we would like to establish a more general recommendation model to capture the reliabilities of users, items and user-item ratings utilizing the techniques in the research field of outlier detection and recommendation systems. We plan to construct a statistical model to describe the reliabilities of each user-item rating. The personalized recommendations generated from the new statistical model would be more accuracy and appealing for individual user, as this model emphasizes the importance of the ratings with high dependability and alleviates the adverse effect of inconsistent ratings and shilling attacks.

Bibliography

[1] http://www.cs.umb.edu/ dana/gaclust/index.html.

[2] http://www.datasetgenerator.com/.

[3] http://www.ics.uci.edu/ mlearn/mlrepository.html.

[4] G. Adomavicius and A. Tuzhilin. 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*, 17(6):734–749, June 2005.

[5] C. C. Aggarwal, J. L. Wolf, K.-L. Wu, and P. S. Yu. Horting hatches an egg: A New Graph-Theoretic Approach to Collaborative Filtering. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '99*, Aug. 1999.

[6] C. C. Aggarwal and P. S. Yu. Outlier detection for high dimensional data. In *Proceedings of the 2001 ACM SIGMOD international conference on Management of data, SIGMOD '01*, pages 37–46, 2001.

[7] R. Agrawal, T. Imieliński, and A. Swami. Mining association rules between sets of items in large databases. *SIGMOD Rec.*, 22(2):207–216, June 1993.

[8] E. Aleskerov, B. Freisleben, and B. Rao. Cardwatch: a neural network based database mining system for credit card fraud detection. In *Proceedings of Computational Intelligence for Financial Engineering*, pages 220–226, Mar 1997.

[9] X. Amatriain, J. M. Pujol, N. Tintarev, and N. Oliver. Rate it again: increasing recommendation accuracy by user re-rating. In *Proceedings of the third ACM conference on Recommender systems, RecSys '09*, pages 173–180, 2009.

[10] F. Angiulli, S. Basta, and C. Pizzuti. Distance-based detection and prediction of outliers. *IEEE Transactions on Knowledge and Data Engineering*, 18(2):145–160, Feb. 2006.

[11] F. Angiulli and C. Pizzuti. Outlier mining in large high-dimensional data sets. *IEEE Transactions on Knowledge and Data Engineering*, 17(2):203–215, Feb. 2005.

[12] M. Balabanović and Y. Shoham. Fab: content-based, collaborative recommendation. *Communications of the ACM*, 40(3):66–72, Mar. 1997.

[13] D. Barbará, C. Domeniconi, and J. P. Rogers. Detecting outliers using transduction and statistical testing. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '06*, pages 55–64, 2006.

[14] J. Basilico and T. Hofmann. Unifying collaborative and content-based filtering. In *Twenty-first international conference on Machine learning, ICML '04*, July 2004.

[15] S. D. Bay and M. Schwabacher. Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '03*, pages 29–38, 2003. ACM.

[16] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. *Proceedings of the 14th conference on Uncertainty in Artificial Intelligence*, 461(8):43–52, 1998.

[17] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data, SIGMOD '00,* pages 93–104, 2000.

[18] R. Burke. Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12:331–370, 2002.

[19] R. Burke. Hybrid web recommender systems. *The adaptive web*, pages 377–408, Jan. 2007.

[20] F. Cacheda, V. Carneiro, D. Fernández, and V. Formoso. Comparison of collaborative filtering algorithms. *ACM Transactions on the Web*, 5(1):1–33, Feb. 2011.

[21] P. K. Chan, M. V. Mahoney, and M. H. Arshad. A machine learning approach to anomaly detection. *Technical Report, Florida Institute of Technology*, 2003.

[22] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM Computing Surveys*, 41(3):15:1–15:58, July 2009.

[23] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection for discrete sequences: A survey. *IEEE Transactions on Knowledge and Data Engineering*, 24(5):823–839, May 2012.

[24] M. Chen and J. P. Singh. Computing and using reputations for internet ratings. In *Proceedings of the 3rd ACM conference on Electronic Commerce, EC '01*, pages 154–162, Oct. 2001.

[25] T. M. Cover and J. A. Thomas. *Elements of information theory.* Wiley-Interscience, 1991.

[26] P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the fourth ACM conference on Recommender systems, RecSys '10*, Sept. 2010.

[27] A. S. Das, M. Datar, A. Garg, and S. Rajaram. Google news personalization. In *Proceedings of the 16th international conference on World Wide Web, WWW '07*, May 2007.

[28] K. Das and J. Schneider. Detecting anomalous records in categorical datasets. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '07*, pages 220–229, 2007.

[29] K. Das, J. Schneider, and D. B. Neill. Anomaly pattern detection in categorical datasets. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '08*, pages 169–176, 2008.

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

[31] G. Di Battista, P. Eades, R. Tamassia, and I. G. Tollis. Algorithms for drawing graphs: an annotated bibliography. *Computational Geometry: Theory and Applications*, 4(5):235–282, Oct. 1994.

[32] T. G. Dietterich. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation*, 10(7):1895–1923, Oct. 1998.

[33] J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In *ICML '05*, 2005.

[34] H. J. Escalante. A comparison of outlier detection algorithms for machine learning. In *Proceedings of the International Conference on Communications in Computing*, 2005.

[35] M. Filippone and G. Sanguinetti. Information theoretic novelty detection. *Pattern Recognition*, 43(3):805–814, Mar. 2010.

[36] P. W. Foltz and S. T. Dumais. Personalized information delivery: an analysis of information filtering methods. *Communications of the ACM*, 35(12):51–60, Dec. 1992.

[37] M. Fox, G. Gramajo, A. Koufakou, and M. Georgiopoulos. Detecting outliers in categorical data sets using non-derivable itemsets. *Technical Report, The AMALTHEA REU Program*, 2008.

[38] Frank E. Grubbs. Procedures for Detecting Outlying Observations in Samples. *Technometrics*, 11(1), 1969.

[39] S. R. Gaddam, V. V. Phoha, and K. S. Balagani. K-means+id3: A novel method for supervised anomaly detection by cascading k-means clustering and id3 decision tree learning methods. *IEEE Transactions on Knowledge and Data Engineering*, 19(3):345–354, Mar. 2007.

[40] J. Gao, H. Cheng, and P.-N. Tan. Semi-supervised outlier detection. In *Proceedings of the 2006 ACM symposium on Applied computing, SAC '06*, pages 635–636, 2006. ACM.

[41] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Communications of the ACM*, 35(12):61–70, Dec. 1992.

[42] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A Constant Time Collaborative Filtering Algorithm. *Information Retrieval*, 4(2):133–151, July 2001.

[43] J. Han and M. Kamber. *Data Mining - Concepts and Techniques.* Elsevier, 2006.

[44] F. Harper, X. Li, Y. Chen, J. Konstan, L. Ardissono, P. Brna, and A. Mitrovic. *User Modeling 2005*, volume 3538 of *Lecture Notes in Computer Science.* Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[45] D. M. Hawkins. *Identification of outliers.* Taylor & Francis, 1980.

[46] Q. He, J. Pei, D. Kifer, P. Mitra, and L. Giles. Context-aware citation recommendation. In *Proceedings of the 19th international conference on World Wide Web, WWW '10*, Apr. 2010.

[47] Z. He, X. Xu, and S. Deng. Discovering cluster-based local outliers. *Pattern Recognition Letters*, 24(9-10):1641–1650, June 2003.

[48] Z. He, X. Xu, and S. Deng. An optimization model for outlier detection in categorical data. *ICIC '05*, 2005.

[49] Z. He, X. XU, Z. HUANG, and S. DENG. Fp-outlier: Frequent pattern based outlier detection. *Computer Science and Information Systems*, 2:103–118, 2005.

[50] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '99*, pages 230–237, 1999.

[51] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1):5–53, Jan. 2004.

[52] V. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, Oct. 2004.

[53] T. Hofmann. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*, 22(1):89–115, Jan. 2004.

[54] Y. Hu, Y. Koren, and C. Volinsky. Collaborative Filtering for Implicit Feedback Datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, Dec. 2008.

[55] Z. Huang, H. Chen, and D. Zeng. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems*, 22(1):116–142, Jan. 2004.

[56] N. Hurley and M. Zhang. Novelty and Diversity in Top-N Recommendation – Analysis and Evaluation. *ACM Transactions on Internet Technology*, 10(4):1–30, Mar. 2011.

[57] L. Itti and P. Baldi. Bayesian surprise attracts human attention. In *NIPS '05*, 2005.

[58] M. Jamali and M. Ester. TrustWalker: a random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09*, 2009.

[59] W. Jin, A. K. H. Tung, and J. Han. Mining top-n local outliers in large databases. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '01*, pages 293–298, 2001.

[60] W. Jin, A. K. H. Tung, J. Han, and W. Wang. Ranking outliers using symmetric neighborhood relationship. In *Proceedings of the 10th Pacific-Asia conference on Advances in Knowledge Discovery and Data Mining, PAKDD '06*, pages 577–593, 2006.

[61] G. Karypis. Evaluation of Item-Based Top- N Recommendation Algorithms. In *Proceedings of the tenth international conference on Information and knowledge management, CIKM '01*, page 247, Oct. 2001.

[62] J. F. Kenney and E. S. Keeping. *Mathematics of statistics.* Van Nostrand, 1954.

[63] H.-N. Kim, A.-T. Ji, H.-J. Kim, and G.-S. Jo. Error-based collaborative filtering algorithm for top-N recommendation. In *APWeb/WAIM '07* pages 594–605, June 2007.

[64] E. M. Knorr and R. T. Ng. Algorithms for mining distance-based outliers in large datasets. In *Proceedings of the 24rd International Conference on Very Large Data Bases, VLDB '98*, pages 392–403, 1998.

[65] Y. Koren. Factorization meets the neighborhood. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '08*, 2008.

[66] Y. Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89, Apr. 2010.

[67] H. P. Kriegel, P. Kroger, and A. Zimek. Outlier detection techniques. In *SDM Tutorial '10*, 2010.

[68] Y. Kwon. Improving top-n recommendation techniques using rating variance. In *Proceedings of the 2008 ACM conference on Recommender systems - RecSys '08*, Oct. 2008.

[69] S. K. Lam and J. Riedl. Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web, WWW '04*, pages 393–402, 2004.

[70] P. Lamere and O. Celma. Music Recommendation Tutorial. In *Proceedings of the 8th International Conference on Music Information Retrieval, ISMIR '07*, 2007.

[71] T. Leckie and A. Yasinsac. Metadata for anomaly-based security protocol attack deduction. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1157–1168, Sept. 2004.

[72] W. Lee and D. Xiang. Information-theoretic measures for anomaly detection. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy, SP '01*, 2001.

[73] B. Li, Q. Yang, and X. Xue. Transfer learning for collaborative filtering via a rating-matrix generative model. In *Proceedings of the 26th Annual International Conference on Machine Learning, ICML '09*, pages 1–8, June 2009.

[74] L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web, WWW '10*, Apr. 2010.

[75] S. Li, R. Lee, and S.-D. Lang. Mining distance-based outliers from categorical data. In *Proceedings of the Seventh IEEE International Conference on Data Mining Workshops, ICDMW '07*, pages 225–230, 2007.

[76] S.-d. Lin and H. Chalupsky. Discovering and explaining abnormal nodes in semantic graphs. *IEEE Transactions on Knowledge and Data Engineering*, 20(8):1039–1052, Aug. 2008.

[77] G. Linden, B. Smith, and J. York. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, Jan. 2003.

[78] M. Lipczak and E. Milios. Efficient Tag Recommendation for Real-Life Data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(1):2, Oct. 2011.

[79] T.-Y. Liu. Learning to Rank for Information Retrieval. In *Foundations and Trends' in Information Retrieval*, volume 3, pages 225–331, Mar. 2007.

[80] P. Lops, M. Gemmis, G. Semeraro, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. *Recommender Systems Handbook.* Springer US, Boston, MA, 2011.

[81] Y. Lv, T. Moon, P. Kolari, Z. Zheng, X. Wang, and Y. Chang. Learning to model relatedness for news recommendation. In *Proceedings of the 20th international conference on World Wide Web, WWW '11*, Mar. 2011.

[82] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining, WSDM '11*, Feb. 2011.

[83] L. W. Mackey, D. Weiss, and M. I. Jordan. Mixed Membership Matrix Factorization. In *Proceedings of the 27th International Conference on Machine Learning, ICML '10*, pages 711–718, June 2010.

[84] B. M. Marlin, R. S. Zemel, S. T. Roweis, and M. Slaney. Recommender systems: Missing data and statistical model estimation. In *IJCAI' 11*, 2011.

[85] P. Massa, P. Avesani, and J. Golbeck. *Computing with Social Trust.* Human-Computer Interaction Series. Springer London, London, 2009.

[86] M. R. McLaughlin and J. L. Herlocker. A collaborative filtering algorithm and evaluation metric that accurately model the user experience. In *Proceedings of the 27th annual international conference on Research and development in information retrieval, SIGIR '04*, July 2004.

[87] A. K. Menon, K.-P. Chitrapura, S. Garg, D. Agarwal, and N. Kota. Response prediction using collaborative filtering with hierarchies and side-information. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '11*, Aug. 2011.

[88] T. Miranda, M. Claypool, A. Gokhale, T. Mir, P. Murnikov, D. Netes, and M. Sartin. Combining Content-Based and Collaborative Filters in an Online Newspaper. In *Processding of ACM SIGIR Workshop on Recommendation Systems*, 1999.

[89] H. D. K. Moonesignhe and P.-N. Tan. Outlier detection using random walks. In *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence, ICTAI '06*, pages 532–539, 2006.

[90] R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries, DL '00*, pages 195–204, June 2000.

[91] D. S. Moore, G. P. McCabe, and B. A. Craig. *Introduction to the Practice of Statistics*. W. H. Freeman, 2010.

[92] M. E. Otey, A. Ghoting, and S. Parthasarathy. Fast distributed outlier detection in mixed-attribute data sets. *Data Mining and Knowledge Discovery*, 12(2-3):203–228, May 2006.

[93] S. Papadimitriou, H. Kitagawa, P. Gibbons, and C. Faloutsos. Loci: fast outlier detection using the local correlation integral. In *Proceedings of 19th International Conference on Data Engineering, ICDE '03*, pages 315–326, Mar. 2003.

[94] S.-T. Park and W. Chu. Pairwise preference regression for cold-start recommendation. In *Proceedings of the third ACM conference on Recommender systems, RecSys '09*, Oct. 2009.

[95] M. Pazzani, D. Billsus, P. Brusilovsky, A. Kobsa, and W. Nejdl. *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, 2007.

[96] J. Peng, D. D. Zeng, and Z. Huang. Latent subject-centered modeling of collaborative tagging: An application in social search. *ACM Transactions on Management Information Systems (TMIS)*, 2(3):15, Oct. 2011.

[97] A. Popescul, L. H. Ungar, D. M. Pennock, and S. Lawrence. Probabilistic Models for Unified Collaborative and Content-Based Recommendation in Sparse-Data Environments. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, UAI '01*, pages 437–444, Aug. 2001.

[98] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. *ACM SIGMOD Record*, 29(2):427–438, June 2000.

[99] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Proceedings of the Twenty-Fifth Conference Annual Conference on Uncertainty in Artificial Intelligence, UAI '09*, pages 452–461, June 2009.

[100] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized Markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World Wide Web, WWW '10*, Apr. 2010.

[101] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. GroupLens: an open architecture for collaborative filtering of netnews. In *Proceedings of the 1994 ACM conference on Computer supported cooperative work, CSCW '94*, pages 175–186, Oct. 1994.

[102] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor. Recommender Systems Handbook. *Springer*, 2011.

[103] V. Robu, H. Halpin, and H. Shepherd. Emergence of consensus and shared vocabularies in collaborative tagging systems. *ACM Transactions on the Web*, 3(4):1–34, Sept. 2009.

[104] R. Salakhutdinov and A. Mnih. Probabilistic Matrix Factorization. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20, NIPS '*, 2008.

[105] R. Salakhutdinov, A. Mnih, and G. Hinton. Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning, ICML '07*, pages 791–798, June 2007.

[106] R. Salakhutdinov and N. Srebro. Collaborative Filtering in a Non-Uniform World: Learning with the Weighted Trace Norm. In *Advances in Neural Information Processing Systems 23*, pages 2056–2064, 2010.

[107] B. Sarwar, G. Karypis, J. Konstan, and J. Reidl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the tenth international conference on World Wide Web, WWW '01*, pages 285–295, Apr. 2001.

[108] B. Schölkopf, J. C. Platt, J. C. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13(7):1443–1471, July 2001.

[109] U. Shardanand and P. Maes. Social information filtering: Algorithms for Automating "Word of Mouth". In *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '95*, pages 210–217, May 1995.

[110] X. Song, M. Wu, C. Jermaine, and S. Ranka. Conditional anomaly detection. *IEEE Transactions on Knowledge and Data Engineering*, 19(5):631–645, May 2007.

[111] Y. Song, L. Zhang, and C. L. Giles. Automatic tag recommendation algorithms for social recommender systems. *ACM Transactions on the Web*, 5(1):1–31, Feb. 2011.

[112] E. I. Sparling and S. Sen. Rating: How Difficult is It? In *Proceedings of the fifth ACM conference on Recommender systems, RecSys '11*, Oct. 2011.

[113] S. Srinivasa. A review on multivariate mutual information. *University of Notre Dame*, 2008.

[114] R. E. Steuer. Multiple Criteria Optimization: Theory, Computation, and Application. *Krieger Publishing Company*, 1989.

[115] I. Szpektor, A. Gionis, and Y. Maarek. Improving recommendation for long-tail queries via templates. In *Proceedings of the 20th international conference on World Wide Web, WWW '11*, Mar. 2011.

[116] G. Takács, I. Pilászy, B. Németh, and D. Tikk. Major components of the gravity recommendation system. *ACM SIGKDD Explorations Newsletter*, 9(2):80, Dec. 2007.

[117] J.-i. Takeuchi and K. Yamanishi. A unifying framework for detecting outliers and change points from time series. *IEEE Transactions on Knowledge and Data Engineering*, 18(4):482–492, Apr. 2006.

[118] S. Tan, J. Bu, C. Chen, B. Xu, C. Wang, and X. He. Using rich social media information for music recommendation via hypergraph model. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 7S(1):1–22, Oct. 2011.

[119] D. M. Tax and R. P. Duin. Support vector domain description. *Pattern Recognition Letters*, 20(11-13):1191–1199, 1999.

[120] A. Umyarov and A. Tuzhilin. Improving Collaborative Filtering Recommendations Using External Data. In *2008 Eighth IEEE International Conference on Data Mining*, pages 618–627, Dec. 2008.

[121] A. Umyarov and A. Tuzhilin. Using external aggregate ratings for improving individual recommendations. *ACM Transactions on the Web*, 5(1):1–40, Feb. 2011.

[122] V. Vasuki, N. Natarajan, Z. Lu, B. Savas, and I. Dhillon. Scalable Affiliation Recommendation using Auxiliary Networks. *ACM Transactions on Intelligent Systems and Technology*, 3(1):3, Oct. 2011.

[123] Vic Barnett and Toby Lewis. Outliers in Statistical Data. *John Wiley & Sons*, 1994.

[124] Ville Hautamaki, Ismo Karkkainen, and Pasi Franti. Outlier detection using k-nearest neighbour graph. In *Proceedings of the 17th International Conference on Pattern Recognition, ICPR '04*, 2004.

[125] J. Wang, A. P. de Vries, and M. J. T. Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '06*, Aug. 2006.

[126] J. Wang, A. P. de Vries, and M. J. T. Reinders. Unified relevance models for rating prediction in collaborative filtering. *ACM Transactions on Information Systems*, 26(3):1–42, June 2008.

[127] X. Wang and I. Davidson. Discovering contexts and contextual outliers using random walks in graphs. In *Proceedings of the 2009 Ninth IEEE International Conference on Data Mining, ICDM '09*, pages 1034–1039, 2009.

[128] S. Watanabe. Information theoretical analysis of multivariate correlation. *IBM Journal of Research and Development*, 4(1):66–82, Jan. 1960.

[129] L. Wei, W. Qian, A. Zhou, W. Jin, and J. X. Yu. Hot: hypergraph-based outlier test for categorical data. In *Proceedings of the 7th Pacific-Asia conference on Advances in knowledge discovery and data mining, PAKDD '03*, pages 399–410, 2003.

[130] X. Xin, M. R. Lyu, and I. King. CMAP: effective fusion of quality and relevance for multi-criteria recommendation. In *Proceedings of the fourth ACM international conference on Web search and data mining, WSDM '11*, Feb. 2011.

[131] S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike: joint friendship and interest propagation in social networks. In *Proceedings of the 20th international conference on World wide web, WWW '11*, Mar. 2011.

[132] J. X. Yu, W. Qian, H. Lu, and A. Zhou. Finding centric local outliers in categorical/numerical spaces. *Knowledge and Information Systems*, 9(3):309–338, Mar. 2006.

[133] Q. Yuan, L. Chen, and S. Zhao. Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation. In *Proceedings of the fifth ACM conference on Recommender systems, RecSys '11*, Oct. 2011.

[134] D. Zhang, D. Gatica-Perez, S. Bengio, and I. McCowan. Semi-supervised adapted hmms for unusual event detection. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR '05*, pages 611–618, 2005.

[135] Y. Zheng and X. Xie. Learning travel recommendations from user-generated GPS traces. *ACM Transactions on Intelligent Systems and Technology*, 2(1):2, Jan. 2011.

[136] X. Zhu, J. Guo, X. Cheng, P. Du, and H.-W. Shen. A unified framework for recommending diverse and relevant queries. In *Proceedings of the 20th international conference on World Wide Web, WWW '11*, Mar. 2011.