UNIVERISTY OF PIRAEUS - DEPARTMENT OF INFORMATICS

ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ – ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

MSc «ADVANCED COMPUTING AND INFORMATICS SYSTEMS»

ΠΜΣ «ΠΡΟΗΓΜΕΝΑ ΣΥΣΤΗΜΑΤΑ ΠΛΗΡΟΦΟΡΙΚΗΣ»

MSc Thesis

Μεταπτυχιακή Διατριβή

Thesis Title: Τίτλος Διατριβής:	**Collaborative Filtering Algorithms, Methods and Techniques** Αλγόριθμοι Συνεργατικού Φιλτραρίσματος, Μέθοδοι και Τεχνικές
Student's name-surname: Ονοματεπώνυμο φοιτητή:	**Aikaterini Moustaka** Αικατερίνη Μουστάκα
Father's name: Πατρώνυμο:	**Georgios** Γεώργιος
Student's ID No: Αριθμός Μητρώου:	ΜΠΣΠ15055
Supervisor: Επιβλέπων:	**Georgios Tsihrintzis, Professor** Γεώργιος Τσιχριντζής, Καθηγητής

September 2022 / Σεπτέμβριος 2022

3-Member Examination Committee

Τριμελής Εξεταστική Επιτροπή

Georgios Tsihrintzis	**Efthimios Alepis**	**Dionisios Sotiropoulos**
Professor	**Associate Professor**	**Assistant Professor**
Γεώργιος Τσιχριντζής	Ευθύμιος Αλέπης	Διονύσιος Σωτηρόπουλος
Καθηγητής	Αναπληρωτής Καθηγητής	Επίκουρος Καθηγητής

Περίληψη

Η τεχνολογία αποτελεί πλεόν αναπόσπαστο κομμάτι της καθημερινότητάς μας, καθώς η χρήση των ηλεκτρονικών συσκευών και του διαδικτύου είναι απαραίτητη για την κάλυψη των αναγκών μας. Οι χρήστες χρησιμοποιουν εφαρμογές διαδικτύου και κινητών συσκευών τόσο για τις αγορές όσο και την ενημέρωση και εκπαίδευσή τους. Συνεπώς, η παροχή εξατομικευμένων πληροφοριών, αποτελεί το πλέον επιθυμητό χαρακτηριστικό για τον κάθε χρήστη. Έχοντας ως βάση τις δυνατότητές που προσφέρει η Τεχνητή Νοημοσύνη, έχουν αναπτυχθεί αλγόριθμοι φιλτραρίσματος, οι οποίοι παρέχουν αποτελέσματα εξατομικευμένα προς τον χρήστη.

Σκοπός της παρούσας εργασίας είναι η μελέτη των αλγορίθμων φιλτραρίσματος. Τα χαρακτηριστικά τους, οι τεχνικές καθώς και οι αδυναμίες τους θα αναλυθούν έτσι ώστε, να γίνουν κατανοητές οι δυνατότητές τους και να ξεχωρίσουν εκείνοι που παρέχουν τα καλύτερα αποτελέσματα με τις λιγότερες απώλειες.

Abstract

Technology is now an integral part of our daily life, as the use of electronic devices and the internet is essential to meet our needs. The users use web and mobile applications for both shopping and their information and education. Therefore, the provision of personalized information is the most desirable feature for each user. Taking into consideration, the power of the Artificial Intelligence, collaborative filtering algorithms have been developed, in order for personalized result to be produced for the user.

The purpose of this project, is to study the collaborative filtering algorithms. Their characteristics, techniques as well as their weaknesses will be analyzed in order to understand their potential and distinguish those that provide the best results with the least loss.

Contents

Introduction

In our time, technology is constantly developing. Computers, smartphones and other devices, become an integral part of our daily life. New applications, or new features for existing apps or programs, are developing to help the users complete their tasks easier and in much less time.

Artificial Intelligence has an important role in it. All the algorithms, which are used for filtering and will help the applications and websites to provide the most suitable results per user in each case, are based on it. Some of the most widely used algorithms include collaborative filtering which is mostly used for filtering out items that a user might like on the basis of reactions by similar users, clustering algorithms like k-means, that is useful for efficient, and accurate user segmentation.

The purpose of this project is to study the algorithms that are used to produce more personalized results, for the users and how they are affecting our world. Using the algorithms mentioned above, we will develop an application which, along with the usage of Foursquare API, will aim to provide personalized results for stores and accommodation.

What is Machine Learning?

Machine Learning is a subfield of artificial intelligence, which is broadly defined as a capability of machines to imitate intelligent human behavior. Artificial Intelligence systems are used to perform complex tasks in a way that is similar to how humans solve a problem.

The machine learning algorithms build a model based on simple data, known as training data - the information the machine learning model will be trained on, in order to make predictions or recommendations, without being programmed to do so. They are used in a wide variety of applications, such as email filtering, speech recognition and computer vision where it is difficult to develop algorithms to perform the needed tasks.

A subset of machine learning is related to computational statics, which focuses on making predictions using computers, but not all machine learning is statistical learning. The study of mathematical optimization delivers methods, theory and application domain to the field of machine learning. Data mining is a related field of study, focusing on exploratory data analysis throughout unsupervised learning. In some machine learning implementations, data and neural networks will be used in a way a biological brain would work. In each application across business problems, machine learning is also referred to as predictive analytics.

Data Mining

Machine learning starts with data, numbers, photos or texts like bank transactions, time series from sensors, sales records, etc. The data will be gathered and formulated in order to be used. The more data, the better the program.

From there, programmers choose a machine learning model to use, supply the data and let the computer model to train itself to learn patterns or make predictions. Over time, the human programmer can tweak the model, including changing its parameters, to help push it to more accurate results.

In order to test the accuracy of the machine learning model, when the new data are produced, some of the training data will be held out, to be used as evaluation data. The produced model will be used with different sets of data in the future.

Machine learning and data mining employ the same methods and overlap significantly, but while focusing on prediction, based on knowing properties learned from the training data, data mining focuses on the discovery of unknown properties in the data. Many machine learning methods will be used, each for different purposes. On the other hand, machine learning also employs data mining methods, as “unsupervised learning” or as preprocessing steps to improve the learner's accuracy.

Types of Machine Learning

As we have already mentioned, machine learning involves showing a large volume of data to a machine, so it can learn and make predictions, find patterns or classify data. The used algorithm defines the machine learning type, which works a bit differently. There are three learning machine types: supervised, unsupervised and reinforcement learning.

Supervised Learning

This type of machine learning feeds historical input and output data in machine learning algorithms, with processing in between input/output pairs that allows the algorithm to shift the model to create outputs as closely aligned with the desired result as possible. Common algorithms used for supervised learning include neural networks, decision trees, linear regression and support vector machines.

This machine learning type got its name because the machine is “supervised” during the process of learning, which means that the programmer feeds the algorithm with information, to help it learn. The outcome that is provided to the machine is labeled data, and the rest of the information that is passed, is used as input features.

Supervised learning is effective for a variety of business purposes, including sales forecasting, inventory optimization and fraud detection. Some example of use cases include:

- Predict real estate prices
- Classify whether bank transactions are fraudulent or not
- Finding disease risk factors
- Determining whether loan applicants are low-risk or high-risk
- Predicting the failure of industrial equipment’s mechanical parts

Unsupervised Learning

Unsupervised learning is used in order to analyze and cluster unlabeled datasets. In contrast with supervised learning these algorithms attempt to discover hidden patterns, without the help of a user. They are useful for identifying patterns and using data to make decisions.

Some common use cases include:

- Customer group creation based on purchase behavior
- Grouping inventory according to sales and/or manufacturing matrix
- Identifying associations in customer data

Semi-supervised Learning

To overcome the disadvantages of supervised and unsupervised learning, semi-supervised learning is developed. It consists of a mixture of labeled and unlabeled data, with the latest to overpower the labeled ones.

In this machine learning type, the algorithms use the labeled data, in order to find the unlabeled in them. There are some sources that are characterized by noise or limit, which are called weak supervision. They are much more approachable due to their low price resulting in larger effective training sets.

Reinforcement Learning

Reinforcement learning is a machine learning type that is the closed to how human beings learn in their lives. It uses a feedback technique. The algorithm, acquired a knowledge by interacting with its environment and getting positive or negative rewards for the decisions it makes. Some of the algorithms are Q-learning and deep adversarial networks.

Because this type of machine requires less management than supervised learning, it is viewed as easier to work with dealing with unlabeled data sets.

Some examples of use include:

- Automated parking of cars
- Dynamically controlling traffic lights

Machine Learning Models

To perform machine learning, a model must be created. It has to be trained with some training data in order to be able to make predictions by processing additional data in the future. For machine learning systems, several types of models have been researched.

Artificial Neural Networks

Artificial neural networks, or connectionist systems, are computing systems vaguely inspired by the biological neural networks that constitute animal brains. Systems like these master to perform tasks by taking into consideration cases, mostly without being programmed with any task specific rule.

An Artificial neural network is a model based on a collection of connected nodes called "artificial neurons", which loosely model the neurons in a biological brain. They exchange information via signals. Each artificial neuron that receives it, will process it and then pass it to the next one that is connected with it. Likewise, with Artificial Neural Network implementations, the signal at a connection between artificial neurons is a real number, and the output of each artificial neuron is computed by some non-linear function of the sum of its inputs. The connections between artificial neurons are called "edges". Artificial neurons and edges have a weight which adjusts as the learning proceeds. This weight affects the strength of the signal, at a connection. Artificial neurons have a threshold in order that the signal will be sent only if the aggregate signal crosses that threshold. Typically, artificial neurons are aggregated into layers. Different layers could execute different kinds of transformations on their inputs. Signals travel from the first layer to the last, possibly after traversing the layers multiple times.

The original purpose of the artificial neural network was to resolve problems identical to the human brain. However, inch by inch, the focus moved to performing specific tasks and then to deviations from biology. Some of the tasks that artificial neural networks have been used include video games, social network filtering, and speech recognition.

Decision Trees

Decision tree learning uses a predictive model in order to observe an item and come to a conclusion about its value. This approach is often used in statistics, data mining and machine learning.

The models the target variable can take a discrete set of values are known as classification trees. In these structures, the leaves represent the class labels and the branches the conjunctions of features that lead to them.

Regression trees are when the selected variable can take continuous values. In data mining, decision trees describe data, but the decision making will be made by the result of the classification tree.

Support-vector machines

Support-vector machines, also known as support-vector networks, are a set of related supervised learning methods used for classification and regression. Given a set of training examples, each marked as belonging to one of two categories, a support-vector machine training algorithm builds a model that predicts whether a new example falls into one category or the other.

A support-vector machine training algorithm is a non-probabilistic, binary, linear classifier, although methods such as Platt scaling exist to use support-vector machine in a probabilistic classification setting. In addition to performing linear classification, support-vector machines can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces.

Regression Analysis

Regression analysis encompasses a great collection of statistical methods, in order to estimate the relationship between entry parameters and their associated features.

Linear regression is the most known form, where in order to best fit the given data, according to a mathematical criterion such as ordinary least squares, a single line is drawn. This is frequently extended to mitigate overfitting and bias, just like the ridge regression by regularization methods.

Bayesian networks

A Bayesian network, or directed acyclic graphical model is a probabilistic graphical model that represents a set of random variables and their conditional independence with a directed acyclic graph (DAG). For instance, probabilistic relationships between diseases and symptoms could be represented by a Bayesian network.

Taking into consideration the symptoms, the network will be able to compute the probabilities of the presence of several diseases.

Dynamic Bayesian networks are known as the networks that model sequences of parameters, such as speech signals. Finally, the generalizations of Bayesian networks that can produce and solve, under uncertainty, problems are called influence diagrams.

Genetic algorithms

A genetic algorithm is a search algorithm and heuristic technique that mimics the process of natural selection, using methods such as mutation and crossover to generate new genotypes in order of finding the best solutions to a given problem. The most used algorithms in the 1980s and 1990s were the generic algorithms. Inversely, the machine learning techniques were used in order to improve the performances of the generic and evolutionary algorithms.

Training models

In order to train a model, a large and representative amount of sample data should be collected by the machine learning engineers. The training set data can be as varied as a collection of texts or images and data that have been collected from the individual users of a service.

Overfitting is something to watch out for, in training cases. Biased or non-evaluated data leads to undesired predictions. Bias models may result in detrimental outcomes that could affect the negative impacts on society or objectives. Algorithmic bias is known as the potential result of not fully prepared data for training. Machine learning ethics is becoming a field of study and notably being integrated within machine learning engineering teams.

Federated learning

Federated learning is an artificial intelligence form, which is used in order to train machine learning models which have decentralized the training process. This way, the users are allowed to maintain their privacy by not sending their data to a centralized server. This way, the decentralizing training process of many devices is increased.

Recommendation Engines

Nowadays, many companies use machine learning to make relevant recommendations and grow revenue. For this, data scientists need to choose the best recommendation algorithm among a variety of possible choices according to a business’s limitations and requirements.

What are Recommendation Engines?

In order to provide the most relevant items to users, recommendation engines are used. To be able to provide the best recommendations, they attempt to analyse data by using various algorithms. The products that these algorithms propose are based on the user's past behavior.

Customer-based recommendations can create positive feedback and lead to frequent visits. That is the reason why companies who want to increase their sales, are willing to build intelligent recommendation engines that will be studying the past behavior of their users.

Phases of Recommendation Engines

A typical recommendation engine is based on the following four phases:

- Data collection
- Storage
- Analysis
- Filtering

Data Collection

There are two methods to collect data. We can collect data explicitly or implicitly. Implicit data is information that is not provided intentionally by the user. For example, data streams like clicks, search history, and order history. Explicit data is information that is provided intentionally by the user, like when the user rates a movie.

Data Storage

The quality of a recommendation model will be determined by the amount of data that the algorithm will use. The type of storage will be defined by the type of data. For unstructured data, a NoSQL database will be used or some sort of object storage. SQL databases are also included.

Data Analysis

Data analysis methods are used in order to discover items that are closed to the user's engagement data. To provide immediate user recommendations, intelligent analysis is needed. There are three ways to analyze the data:

- **Real-time systems** - support tools that can analyze and process real-time streams of events.
- **Batch analysis** - this approach requires the programmers to process the data periodically. There must be enough data in order to make the analysis relevant.
- **Near-real-time analysis** - enables the possibility to refresh analyses every few minutes or seconds by quickly collecting data. During the same browsing session, the best recommendations are provided by a near-real-time system.

Data Filtering

In order to generate user recommendations, data filtering is used to find the relevant data. There are some algorithms that are often used, depending on the recommendation engine needs:

- **Content-based** - proposes products which are preferred or viewed by the selected user.
- **Cluster** - recommended products of the same type. For example, the same product with different colors.
- **Collaborative** - recommends products that have been viewed or liked by other users. With the use of this algorithm, it is presumed that two users with similar taste in the past will probably like similar items in the future.

Recommendation and Personalization Algorithms

Clustering Algorithms

Clustering is a part of unsupervised machine learning. It analyzes unlabeled data, divides it into groups with similar characteristics and groups them into clusters. The data that are classified by the clustering algorithms, point into a specific group. Data points with similar features or properties must be in the same group.

The most popular algorithm is k-means. It is used for efficient and accurate user segmentation. Customer segmentation divides customers into groups based on shared characteristics and qualities.

The purpose of the segmentation is to determine which is the best way to approach the customers in each group, in order to optimize conversions. To make the personalization more manageable, customers are grouped together. Then the marketers can use the customer's data to create personalized offerings or deliver customers with hyper-personalized content via personalization platforms.

The basic steps of this algorithms are:

- **Estimating the clusters centroids** - The (K) number of centroids, need to be estimated and defined in advance.
- **Assigning data sets to the nearest centroid** - the assignments must be based on the Euclidean distance.
- **Move the centroid** - the new value of a centroid is going to be the mean of all the others in a cluster.

The algorithm repeats these steps until no more improvements are possible.

Across business fields and industries there are various clustering applications. It can be used for recommendation engine development, social media analysis and other.

Regression Analysis

Regression is a supervised machine learning algorithm that is used in order to define the correlation between a dependent target and an independent variable. This technique is used for:

- Determination the strength of the relationship between variables
- Prediction of the outcomes when independent variables change
- Point estimation

There are many forms of regression. Logistic and linear modeling techniques are the most popular:

- **Linear regression** — it is a statistical analysis that defines the most significant variables for prediction. If it is single or multiple, it will be defined by the number of independent variables.
- **Logistic regression** - based on previous observations of data sets, the data values will be predicted.

Collaborative Filtering

Collaborative Filtering and its variants are some of the most commonly used recommendation algorithms. With this technique, we can filter out items that a user might like on the basis of reactions by similar users.

It searches a large group of people in order to find a smaller set of users with similar tastes as a particular user. It combines the most liked items, in order to create a ranked list of suggestions.

There are two classes of Collaborative Filtering.

- **User-based** - measures the similarity between the selected users and others.
- **Item-based** - measures the similarity between the items that selected users rate or interact with and other items.

Challenges

In order to produce high-quality recommendations, the collaborative filtering algorithms, will have to address the challenges, which are also characteristics of collaborative filtering tasks as well.

The most common challenges are:

- Data Sparsity
- Scalability
- Synonymy
- Gray Sheep
- Shilling Attacks

Data Sparsity

In practice, many commercial recommender systems use large product sets, in order to produce the right results. The user-item matrix that is used will be extremely sparse and the performances of the predictions are challenged.

The data sparsity challenge appears in several situations. Specifically, the *cold start* problem occurs when a new user or item enters the system. It is difficult to find similar ones, thus there is not enough information (lack of raking or purchase history). This is also known as the *new user problem* or the *new item problem*.

Another problem is the *reduced coverage* problem, where coverage is the percentage of items that the algorithm could provide recommendations for. It occurs when the number of users' ratings may be very small compared with the large number of items in the system.

Finally, the *Neighbor transitivity*, which refers to a problem with sparse databases, where users with similar tastes may not be identified as such if they have not both rated any of the same items.

To alleviate the data sparsity problem, many approaches with different techniques have been proposed, like *Singular Value Decomposition* and *Latent Semantic Indexing*.

Singular Value Decomposition

Singular Value Decomposition, is included in dimensional reduction techniques. It removes unrepresentative or insignificant users or items to reduce the dimensionalities of the user-item matrix directly.

Latent Semantic Indexing

Latent Semantic Indexing uses a patent, to retrieve information, that is based on Singular Value Decomposition. Here similarity between users is determined by the representation of the user in the reduced space.

Nevertheless, when certain users or items are discarded, useful information which can be used for recommendations related to them may be lost and the prediction quality may be degraded.

Helpful to address the sparsity problem, are hybrid collaborative filtering algorithms, such as the *content-boosted* collaborative filtering algorithm. These algorithms use external content information for the production of predictions for new users or new items.

Another approach is with model-based collaborative filtering algorithms, like TAN-ELR, which address the sparsity problem by providing more accurate predictions for sparse data. Some of the techniques that are used include the association retrieval technique, which applies an associative retrieval framework and related spreading activation algorithms to explore transitive associations among users through their rating and purchase history.

Scalability

As the numbers of existing users and items grow, traditional collaborative filtering algorithms will suffer with serious scalability problems, with computational resources going beyond practical or acceptable levels. For example, with tens of millions of customers (M) and millions of distinct catalog items (N), the collaborative filtering algorithm with complexity of O(n) is already too large. However, the systems have to react immediately to online requirements and make recommendations for all users, regardless their purchase and rating history. This requires a high scalability of collaborative filtering systems.

There are some approaches that can address the scalability problem. Dimensionality reduction techniques like Singular Value Decomposition and memory based collaborative filtering algorithms are included. The Singular Value Decomposition technique will produce good quality recommendations quickly, but they have to undergo expensive matrix factorization steps.

On the Singular Value Decomposition approach, the decomposition is precomputed, using existing users. When a new set of ratings are added to the database, the *folding-in* projection technique is used, for an incremental system without recomputing the low dimensional model from scratch, to be built. Therefore, this makes the recommender system highly scalable.

On the other hand, memory based collaborative filtering algorithms, such as the item based *Pearson correlation*, can achieve satisfactory scalability. This algorithm, instead of calculating similarities between all pairs of items, it calculates the similarity only between the co-paired items by the user.

Finally, another approach is with model based collaborative filtering algorithms, like the clustering algorithms, which address the scalability problem by seeking users for recommendation within smaller and highly similar clusters instead of the entire database.

Synonymy

The tendency of a number of similar items to have different names or entries is called Synonymy. As a result, that most of the recommender systems are unable to discover this latent association, they treat these products differently. For instance, "movie" and "film", is the same, even though they seem different. In case of a memory based algorithm, it will be impossible to compute similarity as there will be no match.

Therefore, the prevalence of the synonyms decreases the recommendation performance of collaborative filtering systems. The Singular Value Decomposition techniques and particularly the Latent Semantic Indexing method, are capable of dealing with the synonymy problems.

The Singular Value Decomposition takes a large matrix of tem-document association data and constructs a semantic space where terms and documents that are closely associated are placed closely to each other. It allows the arrangement of the space to reflect the major associative patterns in the data and ignore the smaller, less important ones.

However, a partial solution to this problem is given with the use of the Latent Semantic Indexing, which refers to the fact that most words have more than one meaning.

Gray Sheep

Gray Sheep refers to the users whose opinions do not consistently agree or disagree with any group of people and therefore do not benefit from the collaborative filtering. There is also the *Black Sheep* problem. The users who have idiosyncratic tastes, are called Black Sheep. The recommendations and predictions for them are nearly impossible.

Even though it is considered a failure of the recommender system, it is acceptable due to the problems of non-electronic recommenders.

Shilling Attacks

In a system, where everyone can provide recommendations, people have the tendency to give tons of positive recommendations for their own materials and negative ones for their competitors. It is often necessary for collaborative filtering systems to introduce precautions to discourage this kind of manipulation.

Other Challenges

Collaborative filtering systems raise concern about the personal privacy of users as many of them may not want their habits or views to be widely known. Therefore, there are some approaches, which aim to protect users' privacy from recommendation tasks.

Increased noise, known as *sabotage,* is another challenge, as the user population becomes more diverse. Ensembles of maximum margin matrix factorizations and instant selection techniques are found useful to address this problem.

Collaborative Filtering Techniques

There are different types of collaborative filtering:

- Memory based
- Model based
- Hybrid

Below we will analyze the techniques, which are used with each type.

Memory based Collaborative Filtering

Memory based algorithms use a sample or the entire user-item database in order to generate a prediction. Each user is a part of a group of people with similar interests. A prediction for a new user, will be produced by identifying the so-called neighbors of his.

For example, the neighborhood based algorithm, which is a memory based collaborative filtering algorithm, follows the steps below. First, it calculates the similarity or the weight, wi,j, which reflects distance, correlation or weight, between two users or items, i and j. Then it produces a prediction for the active user by taking the average weight of all the ratings of the user or item, for a certain item or user, or by using a simple average weight.

Another example, is when is asked to generate a top-N recommendation, where *k* most similar users or items, needs to be found after computing the similarities and then aggregate the neighbors to get the top-*N* most frequent item as the recommendation.

The steps that the collaborative filtering system follows are:

- Similarity Computation
- Prediction and Recommendation Computation

Similarity Computation

Similarity Computation between users or items, is a critical step in memory based collaborative filtering algorithms. The main point of the similarity computation between two items i and j, for the item based algorithms, is at first to work on the users who have rated both of them and then to apply a similarity computation to determine the similarity of the co-rated items of the users. On the other hand, for the user based algorithms, the similarity for two users who have both rated the same items, is the first to be calculated.

There are many different methods to compute similarity between users or items. Some of them are listed below.

- Correlation Based Similarity
- Vector Based Similarity
- Jaccard Similarity Index

Correlation Based Similarity

Correlation based similarity uses Pearson correlation, in order to measure the similarity between two users or items.

Pearson correlation measures the extent to which two variables are linearly related with each other. In case of a user based algorithm, the below equalization is used.

$$w_{u,v} = \frac{\sum_{i \in I}(r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I}(r_{u,i} - \bar{r}_u)^2}\sqrt{\sum_{i \in I}(r_{v,i} - \bar{r}_v)^2}},$$

Where the i ∈ I sumotions are over the items that both u and v users have rated and $\vec{r}_u$ the average rating of the co-rated items of the uth user.

For the user based algorithms, with u ∈ U, the equation will be like

$$w_{i,j} = \frac{\sum_{u \in U}(r_{u,i} - \bar{r}_i)\left(r_{u,j} - \bar{r}_j\right)}{\sqrt{\sum_{u \in U}(r_{u,i} - \bar{r}_i)^2}\sqrt{\sum_{u \in U}(r_{u,j} - \bar{r}_j)^2}},$$

Where $r_{r,j}$ is the rating of *u* user on item *i* and $\vec{r}_i$ the average rating of the ith item by those users.

Other correlation based algorithms are:

- Constrained Pearson correlation - users medpoint instead of mean rate
- Spearman Rank correlation - in this case the ratings are ranks
- Kendall's *τ* correlation - similar to Spearman, only the relative ranks are used

Vector Cosine Based Similarity

In the Vector Cosine case, the similarity between two documents can be measured by treating each one as a vector of word frequencies and computing the cosine of the angle formed by frequency vectors. This can be adopted in collaborative filtering, which uses items or users instead of documents and ratings rather than frequencies.

If R is the *m* X *n* user-item matrix, then the definition of the similarity of two items *i* and *j*, is the cosine of the *n* dimensional vectors which correspond to the *i*th and *j*th column of matrix R.

For the j and i items, vector cosine similarity is given by

$$w_{i,j} = \cos\left(\vec{i}, \vec{j}\right) = \frac{\vec{i} \bullet \vec{j}}{\|\vec{i}\| * \|\vec{j}\|},$$

Where "•" signify the dot-product of the two vectors.

For example, the vector cosine similarity of $\vec{A}$ = {x_1,y_1} and $\vec{B}$ = {x_2,y_2} is defined as :

$$w_{A,B} = \cos\left(\vec{A}, \vec{B}\right) = \frac{\vec{A} \bullet \vec{B}}{\|\vec{A}\| * \|\vec{B}\|} = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2}\sqrt{x_2^2 + y_2^2}}.$$

Adjusted cosine similarity is used in case the users make use of different rating scales. It subtracts the corresponding user average from each co-rated pair.

Jaccard Similarity Index

Jaccard similarity index, is also known as Jaccard similarity coefficient, is used for comparing the similarity and diversity of sample sets. For users u and v , with n binary attributes each, the Jaccard coefficient measures the degree of overlap between the two sets by dividing the numbers of the observed items for both users (intersection) and the number of different items from both sets of rated items (union).

$$J(u,v) = \frac{|u \cap v|}{|u \cup v|}$$

Jaccard Distance

Jaccard distance measures the dissimilarity of two sets. It can be found, either by subtracting the Jaccard coefficient from 1, $d(u,v) = 1 - J(u,v)$, or by dividing the difference of the sizes of the union and the intersection of two sets by the size of the union $d(u,v) = \frac{|u \cup v| - |u \cap v|}{|u \cup v|}$.

Predictions and Recommendation Computation

The most important step in a collaborative filtering system is to be able to produce predictions or recommendations. In a neighborhood based algorithm, a weighted aggregate of ratings from the nearest users of the active user based on their similarities, will be used to generate predictions.

There are two techniques:

- Weighted Sum of Others' Ratings
- Simple Weighted Average

Weighed Sum of Others' Ratings

To make a prediction to a user *u* for the item *a*, we could take a weighted average of the ratings for the item *a* based of the following equation:

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u) \cdot w_{a,u}}{\sum_{u \in U} | w_{a,u} |},$$

In which $\vec{r}_u$ and $\vec{r}_a$ are the average ratings for the users *a* and *u* and $w_{a,u}$ the weight between the users. The summations occurs from all the users, with u $\in U$ who have rated the item. It is important to notice that the prediction is based on the neighborhood of the active users.

Simple Weighted Average

In case of an item based prediction, we can predict the rating $P_{u,i}$, for user *u* on item *i*, by using the simple weighted average with the equation below :

$$P_{u,i} = \frac{\sum_{n \in N} r_{u,n} w_{i,n}}{\sum_{n \in N} | w_{i,n} |}$$

Where the summations are all the other rated items n $\in N$ for user *u*, $w_{i,n}$ is the weight between the items *n* and *r* and $r_{u,n}$ is the rating for user *u* on item *n*.

Top-N Recommendations

To recommend a set of the Top-N ranked items to a user, Top-N recommendation is used.
The set of items that will be shown will be of interest of the certain user. For example, if the user is a returning customer, the site that has logged in, will recommend a list of items that will be of his or her interest to.

The Top-N recommendation techniques analyze the user-item matrix to discover relations between different users or items that will be used to compute the recommendations. There are two types of algorithms:

- User-Based Top-N Recommendation Algorithm
- Item-Based Top-N Recommendation Algorithm

User-Based Top-N Recommendation Algorithm

The first step in the user-based top-n recommendation algorithm is to identify the *k* most similar users to the active user, also known as nearest neighbors. For this, Pearson correlation or vector-space model will be used. Each user will be treated as a vector in the m-dimensional item space and the similarities between the active user with the other users, will be computed between the vectors.

After the similar users are discovered, the next step is to identify a set of *C* items, which have been purchased from the group, together with their frequency. With the set *C*, the most frequent item that the active user has not yet purchased will be recommended, with the use of user-based collaborative filtering techniques. Unfortunately, the user-based top-N

recommendation algorithms have some limitations related to scalability and real time performance.

Item-Based Top-N Recommendation Algorithms

Item-based Top-N recommendation algorithms have been developed in order to address the scalability problem of user-based top-N recommendation algorithms. Firstly, the *k* most similar items will be computed for each item according to the similarities. Then the set *C* will be defined as candidates of recommended items by taking the union of *k* most similar items and removing each item that is already purchased from the user, in the *U* set. Lastly, the similarities of the items between the sets *U* and *C* will be calculated. The resulting set of items in set *C*, will be the recommended top-N list, sorted in decreasing order of the similarity.

Sadly, the problem after using this method, is that the joint distribution of a set of items is different from the distribution of the individual items in the set and this could potentially produce suboptimal recommendations. To address this problem, a higher-order item based top-N algorithm has been developed by Karypis and Deshpande (*source N.1)*. This algorithm will use all combinations of the items up to a particular size when they are determined to be recommended to a user.

Extensions to Memory-Based Algorithms

There are some extensions that have been developed for the memory based algorithms that are listed below

- Default Voting
- Inverse User Frequency
- Case Amplification
- Imputation-Boosted CF Algorithms
- Weighted Majority Prediction

Default Voting

In some collaborative filters, the similarity is only computed from the ratings in the intersection of the items both users have rated. If the items that they have in common are not that many, their weights tend to be overemphasized. To solve this problem, the default voting simply adds a number of imaginary items that the two users have rated in common, in order to smooth the votes.

Inverse User Frequency

There is an intuition that commonly enjoyed items are less important to weight than rare ones.

The inverse frequency can be defined as

$$f_j = \log (n / n_j)$$

Where n_j is the number of users that have rated the item and n is the total number of users. If all the users have rated the item j, then f_j will be equal to zero. To apply inverse frequency, transformed rating will be used, which is the original rating multiplied by the f_j factor.

Case Amplification

In this case, a transformation will be applied to the weight which was used for the prediction. For the transformation each weight will be amplified by an exponent so that the higher weights get higher or the lower ones get lower.

$$w'_{i,j} = w_{i,j} \cdot \left| w_{i,j} \right|^{\rho-1}$$

Where ρ is the case amplification power. $\rho \geq 1$. The typical choice of $\rho = 2.5$;

Imputation-Boosted Collaborative Filtering Algorithms

In cases where the rating data are extremely sparse and it is very difficult to produce accurate predictions using Pearson correlation based, imputation boosted collaborative filtering is used.

Firstly, this algorithm uses an imputation technique in order to fill in the missing data and then with the help of Pearson correlation it predicts a specific user rating for a specific item.

Imputation Techniques

The most common imputation methods are listed below.

Mean Imputation

By using mean imputation, the mean of the observed values for each variable is computed and the missing values for that variable are imputed by this mean.

It has the advantage of keeping the same mean and the same sample size, but there are many disadvantages such as the reduction of the variance of the imputed variables.

Substitution

With substitution, we could impute the value for a new individual which was not selected to be in the sample. In other words, a new subject will be found and its value will be used instead.

Hot Deck Imputation

In the hot deck imputation case, in order for an imputation to be made, a random value is chosen from an individual in the sample, which has similar values on other variables. To rephrase it, first all subjects which are similar with the other variables will be found and then, one of their values will be chosen randomly, on the missing variable. The advantage of this method is that the results are constrained to only possible values.

Cold Deck Imputation

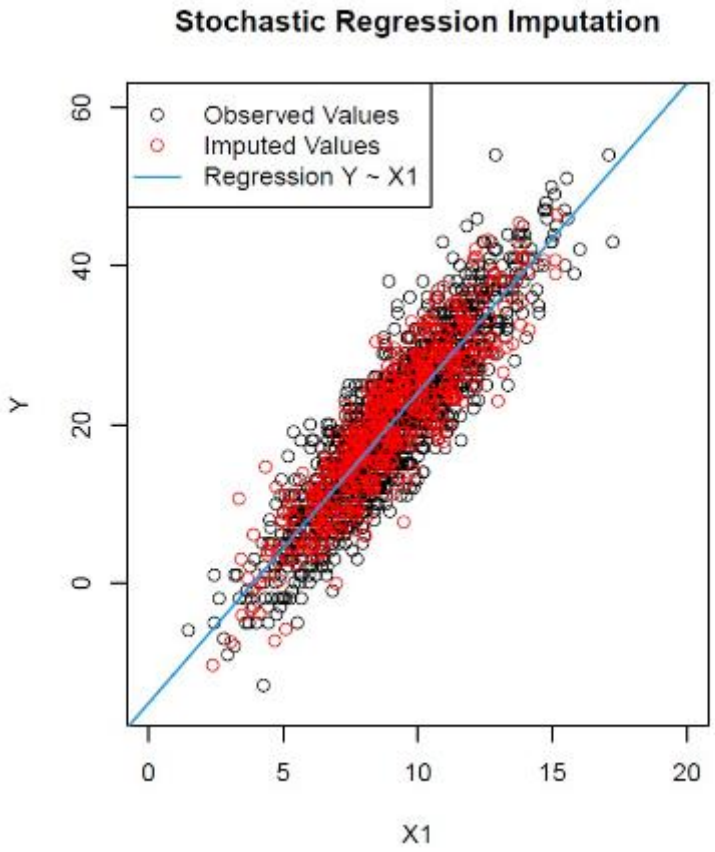
The cold deck imputation refers to using preprocessed data. In other words, data from previous data collection or a different data set.

Regression Imputation

A regression model is estimated to predict observed values of a variable based on other variables, and that model is then used to impute values in cases where the value of that variable is missing.

Stochastic regression Imputation

By using stochastic regression imputation a random error term will be added to the predicted value and therefore the correlation between two variables will be reproduced more appropriately. The graphic below shows the observed values and the imputed values, in an imputation example.



1: Diagram of Stochastic Regression Imputation

Weighted Majority Prediction

The weighted majority prediction algorithm is used to make predictions using the rows with observed data in the same column, weighted by the believed similarity between the rows, with binary rating values. When the compared values are the same, the weights are increased by multiplying it by (2 - γ) and decreased by multiplying by γ when different, with $\gamma \in (0.1)$. This algorithm can be generalized to multiclass data and be extended from user-to-user similarity to item-item similarity and user-item combined similarity (*source n.2*).

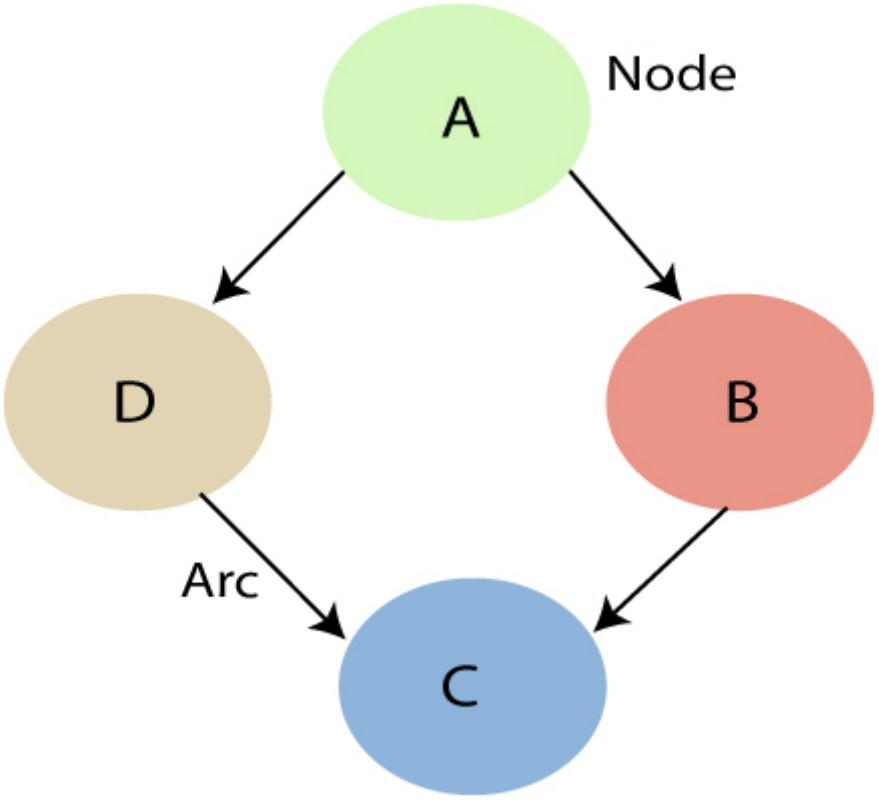
Model-Based Collaborative Filtering

The design and the development of models allows the system to learn to recognise complex patterns based on the training data before proceeding to produce intelligent predictions for the collaborative filtering tasks for the given data, based on the learning methods. Model-based Collaborative filtering algorithms, including Bayesian models, clustering models and dependency networks, have been investigated to solve the shortcomings of memory based collaborative filtering algorithms (*source n.3*), Next are some model based algorithms.

Bayesian Belief Net Collaborative Filtering Algorithms

The Bayesian Belief Net is a directed, acyclic graph with a triplet <*N,A,Θ*>, where each

- node n ∈ *N* represents a random variable
- Arc (also known as directed link) α ∈ *A* between nodes is a probabilistic association between variables and
- *Θ* is the conditional probability table which defines the dependance of the node to its parents



2: Bayesian Belief Net Graph

In the above image, a Bayesian network graph is portrayed. Bayesian belief nets are often used in case of classification tasks, such as classifying if an email is a “spam” or not.

Algorithms

Simple Bayesian Collaborative Filtering Algorithm

For predictions to be made for collaborative filtering tasks, the simple Bayesian algorithm uses a naive Bayes strategy. Assuming that the features, given the class, are independent, the probability of a certain class, given all the features, can be computed. After that, the class with the highest probability will be classified as the predicted class (*sourcen.4*). In case of incomplete data, the probability will be computed over observed data. Same goes for the classification production. The next equalization describes the above.

$$\text{class} = \underset{j \in \text{classSet}}{\arg\max}\, p\left(\text{class}_j\right) \prod_o P\left(X_o = x_o \mid \text{class}_j\right)$$

To smooth the probability calculation and to avoid a conditional probability of 0, *Laplace Estimator* is used :

$$P(X_i = x_i \mid Y = y) = \frac{\#(X_i = x_i, Y = y) + 1}{\#(Y = y) + |X_i|}$$

Where $|Xi|$ is the size of seti $\{Xi\}$.

The simple Bayesian Collaborative Filtering algorithm, because of its in memory calculations for collaborative filtering predictions, can be regarded as a memory based algorithm in comparison to other Bayesian Collaborative Filtering algorithms, which are regarded as model based.

NB- ELR and TAN-ELR Collaborative Algorithms

Extended logistic regression (ELR) is a gradient -ascent algorithm which is a discriminative parameter learning algorithm that maximizes log conditional likelihood.

NB - ELR and TAN - ELR have been proven to have high classification accuracy for both complete and incomplete data. They have better performance compared to simple Bayesian and Pearson correlation algorithms. However, TAN-ELR and NB-ELR need more time to train the models. A solution for this problem is to run the time consuming training stage offline. This way, the prediction producing stage will take less time.

Clustering Collaborative Filtering Algorithms

A cluster is a collection of data that are similar to one another within the same cluster. With the objects that belong to another cluster are dissimilar. The similarity between the objects can be measured using metrics such as *Minkowski* distance and *Pearson correlation*.

The Minkowski distance for two data objects, $X = (x_1,x_2,...,x_n)$ and $Y = (y_1,y_2,...,y_n)$ can be defined using the equalison that follows

$$d(X,Y) = \sqrt[q]{\sum_{i=1}^{n} |x_i - y_i|^q}$$

Where x_i and y_i are the values of the *i*th dimension of object X and Y, n is the dimension number of the object and q is a positive integer. *Manhattan* distance is when q = 1 and *Euclidean* distance is when q = 2.

The clustering methods are classified into three categories:

- Partitioning methods
- Density based methods
- Hierarchical methods

Partitioning Methods

Partitioning clustering methods are used to classify observations within a data set into multiple groups, based on their similarity. The most popular algorithms that are under the partitioning methods are K-Means, PAM (K-Medoid), CLARA.

K-Means Algorithm

K-Means algorithm is used for partitioning the given data sets into k groups, where k represents the number of groups. In K-Means clustering, each cluster is represented by its center, which harmonizes to the mean of points assigned to the cluster.

PAM

PAM stands for “Partition Around Medoids ”. PAM converts its step of PAM from a deterministic computational to a statistical estimation problem and reduces the complexity of a sample size n to O(n log n). Medoids are data points chosen as cluster centers. K-Medoids reduce dissimilarities between points in a cluster and points that are considered as centers of that cluster.

CLARA

Clara, also known as “Clustering Large Applications”, is an extension of K-Medoids. It uses only random samples of the input data and computes the best medoids in those samples. Its performance on crowded datasets is better than K-Medoids.

Density based Methods

Density based clustering methods, typically search for dense clusters for objects, that are separated by sparse regions that represent noise. The most popular methods are DBSCAN and OPTICS.

DBSCAN

DBSCAN is known as Density Based Spatial Clustering of Applications with Noise. It is used in order to discover clusters in different shapes and sizes from a large amount of data, that is containing noise and outliers

This algorithm uses two parameters:

- **minPts**: The minimum number of points that is clustered together for a region to be considered dense. It is defined as ***minPts = 2 * dim***, where dim is the dimension in the dataset. It is important to notice that it may be necessary for larger values to be chosen for very large data or for data that many duplicates are included.

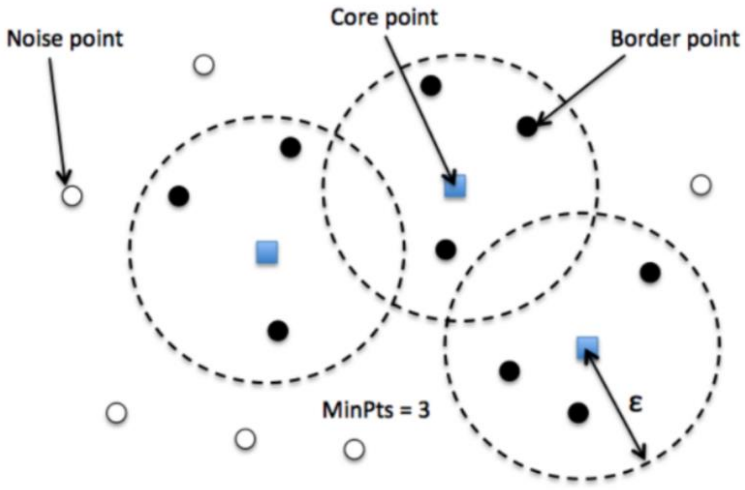
- **Eps (ε)**: It is a distance measure that will be used in order to locate the points in the neighborhood of any point. The value of ***ε***, can be chosen by using a *k-distance graph*, plotting the distance to the ***k = minPts - 1*** nearest neighbor ordered from the largest to the smallest value.

After the completion of the procedure, three types of points are being produced:

- **Core:** This spot has at least *m* points within distance *n* from itself.

- **Border**: This has at least one core point at a distance *n*.

- **Noise:** This is a point which is not a Noise or a Border and has less *m* points within distance *n* from itself.



3: Density Based Spatial Clustering of Applications with Noise

The algorithm follows the below steps:

- Firstly, it proceeds by arbitrarily picking up a point in the dataset.
- If there are at least *minPoint* points within a radius of *ε* to the point, then it is considered all these points to be parts of the same cluster.
- After that, the clusters are expanded by recursively repeating the neighborhood calculation for each neighboring point.

OPTICS

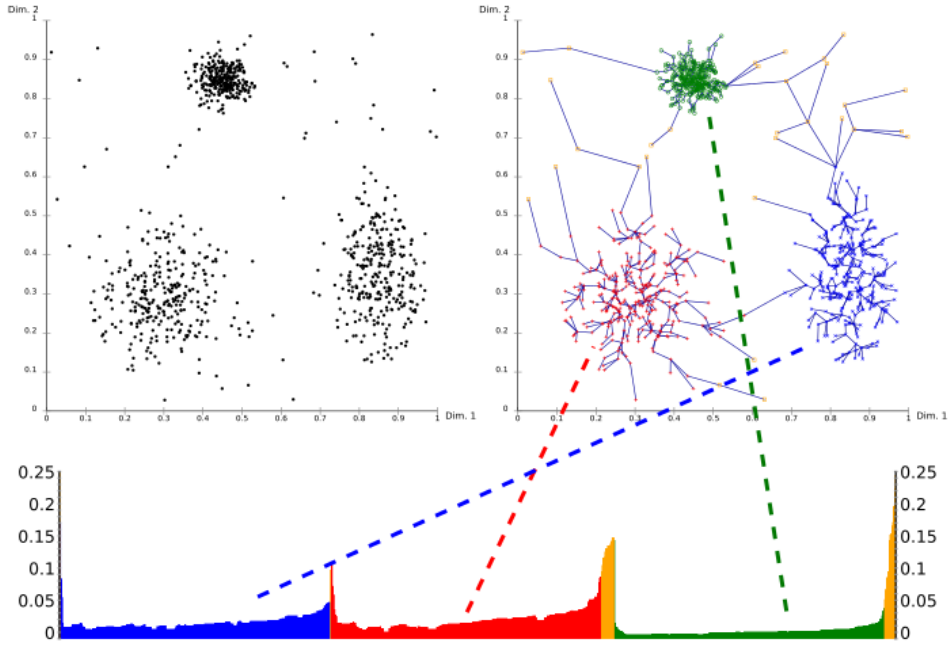
OPTICS is an algorithm that is used for the density based clusters in spatial data to be found. It stands for *Ordering Points To Identify the Clustering Structure* and it was presented by Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel and Jörg Sander. Its basic idea is similar to DBSCAN , but it addresses the latter's major weakness and that is the problem of detecting meaningful clusters in data of varying density.

In order to address the problem, the points of the database are -linearly- ordered in a way that spatially closest points become neighbors in the ordering. Additionally, a special distance is stored for each point that represents the density that must be accepted for a cluster, so that both points belong to the same cluster.

Below, the OPTICS Algorithm is displayed in Pseudocode.

```
OPTICS (SetOfObjects, ε, MinPts, OrderedFile)
  OrderedFile.open();
  FOR i FROM 1 TO SetOfObjects.size DO
    Object := SetOfObjects.get(i);
    IF NOT Object.Processed THEN
      ExpandClusterOrder(SetOfObjects, Object, ε,
                MinPts, OrderedFile)
  OrderedFile.close();
END; // OPTICS
```

The output of the OPTICS algorithm is represented as a *dendrogram*, as the example below, where *reachability-plot* -a special kind of dendrogram - is used.



4: Dendrogram of OPTICS

Hierarchical Methods

Hierarchical Methods are based on a given intercluster distance δ. The most popular is BIRCH, which creates an hierarchical decomposition of the set of data objects using some criterion.

Clustering models have better scalability compared to typical collaborative filtering methods. That is because they make predictions with smaller clusters than the entire base of the customer.

Regression Based Collaborative Filtering Algorithms

Regression algorithms are able to fortel the resulting values according to the input features that are given from the system's data.. A regression method uses an approximation of the ratings to make predictions based on the regression model. If $X = (X_1,X_2,...,X_n)$ to be random variable representing a user's preferences on different items, then the linear regression model will be expressed with the following equalization

$$Y = \Lambda X + N$$

With Λ to be a n X k matrix. $N = (N_1,N_2,...N_n)$ is a random variable representing noise in user choices, Y is an n X m matrix with Y_{ij} is the rating of user i on item j, and finally X is k X m matrix with each column as an estimate of the value of the random variable X for one user. Usually , the matrix Y is very sparse.

MDP-Based Collaborative Filtering Algorithms

MDP stands for “Markov decision process”. It is used for sequential stochastic decision problems, which are often in applications where an agent is influencing its surrounding environment through actions.A MDP can be defined as our four-tuple: {S,A,R,Pr} where

- S is a set of states
- A is a set of actions
- R is an actual value price function for every possible pair of state or action
- Pr has the role of adaptation prospect of pairing the states provided by both actions

The most adequate solution for the MDP is to maximize the function of its reward stream. Firstly, with an initial policy $\pi 0(s) = \arg\max_{a \in A} R(s,a)$, then compute the reward value function $V_i(s)$, which is based on the previous policy and finally update the policy with the new value function. At each step the iteration will converge to an optimal policy.

Latent Semantic Collaborative Filtering Models

Latent semantic technique uses statistical modeling technique which introduces latent class variables in a mixture model setting in order to discover user communities and prototypical interest profiles. To decompose user preferences , user communities have to be overlapped. This technique provides higher accuracy and scalability.

Other Model-Based Collaborative Filtering Techniques

Some model based techniques include a *maximum entropy approach*. This approach clusters the data first and then in a given cluster, uses a maximum entropy model to make predictions.

There is also the dependency network. This is a graphical model for probabilistic relationships, whose graph is potentially cyclic. The probability component in this model is a set of conditional distributions, one for each node given its parents. Dependency networks can provide predictions in a short time and require less time and memory to learn.

Other well known techniques are *decision tree models, Horting, multiple multiplicative factor models, probabilistic principal components analysis*.

Hybrid Collaborative Filtering Techniques

Hybrid Collaborative filtering systems combine collaborative filtering with other recommendation techniques to make predictions or recommendations.

Content-based recommender systems analyze the content of textual information, such as documentations in order to make recommendations. Content based techniques have the *start-up* problem, where they must have enough information to build a reliable classifier. However, there are also limitations on the features explicitly associated with the objects they recommend, compared to collaborative filtering which can make recommendations without any descriptive data. Another problem of the content based techniques, is the overspecialization, in which the only recommendations will be the items that score highly against a user’s profile or their rating history.

Hybrid Recommenders Incorporating Collaborative Filtering and Content Based Features

The content boosted collaborative filtering algorithm uses naive Bayes as the content classifier. In the next step,it fills the rating matrix with the missing values, with the predictions of the content predictor to form a *pseudo* rating matrix. In this pseudo matrix, the observed ratings are kept untouched and missing ratings are replaced by the predictions of a content predictor. Then with the help of a weighted Pearson correlation based collaborative filtering algorithm proceed to make predictions over the resulting pseudo rating matrix. This algorithm gives higher weight for the item that was rated by most users and it also gives higher weight for the active user.

The performance of the content boosted algorithm is more improved compared to some pure content-based recommenders or memory based collaborative filtering algorithms. It is important to note that it overcomes the cold start problem and tackles the sparsity problem of collaborative filtering tasks.

On the next tables, content boosted Collaborative Filtering and its variations are portrayed.

	Content information				Rating matrix				
	Age	Sex	Career	zip	I_1	I_2	I_3	I_4	I_5
U_1	32	F	writer	22904			4		
U_2	27	M	student	10022	2		4	3	
U_3	24	M	engineer	60402		1			
U_4	50	F	other	60804		3	3	3	3
U_5	28	M	educator	85251	1				

5: Content data and originally sparse rating data

Pseudo rating data				
I_1	I_2	I_3	I_4	I_5
2	3	**4**	3	2
2	2	**4**	**3**	2
3	**1**	3	4	3
3	**3**	**3**	**3**	**3**
1	2	4	1	2

***6*: Pseudo data filled by content predictor**

Pearson-CF prediction				
I_1	I_2	I_3	I_4	I_5
2	3	**4**	2	3
3	4	**2**	**2**	3
3	**3**	2	3	3
3	**3**	**3**	**3**	**3**
1	3	1	2	2

7: Predictions from Pearson Collaborative Filtering on pseudo rating data

Hybrid Recommenders Combining Collaborative Filtering and Other Recommender Systems

Using their weights, the weighted hybrid recommender is able to make different recommendation techniques to collaborate with each other. These weights are computed from the results from the available techniques in the system.

The combinations could be linear, the weights adjustable and the weighted average voting or weighted majority voting, as it is also known, can be used.

A switching hybrid recommender switches between recommendation techniques by using some criteria, like confidence levels for recommendation techniques. When there is a problem with the production of a prediction while using a collaborative filtering system, then another recommender system, such as a content based system, will be used. By using the switching hybrid recommenders, the complexity of parameterization of switching criteria will be introduced.

Other hybrid recommenders in this category include:

- Mixed hybrid recommenders
- Cascade hybrid recommenders
- Meta-level recommenders

It is important to note, that the hybrid recommender may produce more accurate recommendations compared to pure collaborative filtering and content-based methods, however, they rely on external information that is not always available. Also, they have increased complexity of implementation.

Hybrid Recommenders Combining Collaborative Filtering Algorithms

To achieve a better performance for the recommendations, the two major classes of Collaborative Filtering, memory based and model based approaches can be combined in order to form a hybrid collaborative filtering approach.

Probabilistic memory-based collaborative filtering

The probabilistic memory based collaborative filtering uses a mixture model built on the basis of a set of stored user profiles. Also, it uses the posterior distribution of user ratings in order to make predictions. For the new user problem to be addressed, an active learning extension to probabilistic memory based collaborative filtering systems is used. When insufficient information is available, an active query is used, for additional user information.

In order to reduce the computation time, the probabilistic memory based collaborative filtering, selects a small subset, which is called *profile space* from the database of user rating and proceeds to generate predictions from that. Compared to Pearson correlation and naive Bayes, the probabilistic memory based collaborative filtering has better accuracy.

Personality diagnosis

Personality diagnosis is a hybrid collaborative filtering approach that combines memory based and model based collaborative filtering algorithms. In this approach, the active user is assumingly generated by an informal choice of another user and adding to its ratings Gaussian noise. In order to calculate the possibility for the active user and the selected user to share the same “personality type”, the given user’s ratings are used. Personality diagnosis can also be regarded as a clustering method, with exactly one user per cluster.

Pennock and Horvitz, propose an associated Personality Diagnosis algorithm. The formula that will be used is defined below

$$\Pr\left(r_i(j) = x \middle| r_i^{\text{true}}(j) = y\right) \propto e^{-(x-y)^2/2\sigma^2}$$

Where the $\mathbf{r}_i^{\mathbf{true}}$ is the mean of an independent normal distribution that is drawn from the user’s *i* actual rating for title *j* and σ is a free parameter.

Evaluation Metrics

The quality of a recommender system is based on the result of the evaluation. The type of metrics that will be used depends on the type of collaborative filtering applications. The metrics evaluating recommendation systems can be classified in the following categories:

- Predictive accuracy metrics
- Classification accuracy metrics
- Rank accuracy metrics
- Normalized distance based performance metric

The most popular metrics are *Mean Absolute Error, Normalized Mean Absolute Error, Root Mean Squared Error and ROC Sensitivity*.

Mean Absolute Error and Normalized Mean Absolute Error

The most used metric in collaborative filtering research literature is Mean Absolute Error.

Mean Absolute Error computes the average of the absolute difference between the predictions and true ratings. It can be defined with the equalization below.

$$MAE = \frac{\sum_{\{i,j\}} \left| p_{i,j} - r_{i,j} \right|}{n}$$

Where *n* is the total number of ratings from all users, $p_{i,j}$ is the predicted rating for user *i* on item *j* and $r_{i,j}$ is the actual rating. The **lower** the Mean Absolute Error is , the **better** the prediction.

Other recommender systems may use different numerical rating scales. *Normalize Mean Absolute Error* normalizes Mean Absolute Error to express errors as percentages of full scale.

$$NMAE = \frac{MAE}{r_{\max} - r_{\min}}$$

Where MAE stands for Mean Absolute Error, r_{max} and r_{min} are the upper and lower bounds of the ratings.

Root Mean Squared Error

Root Mean Squared Error is used by the Netflix prize for movie recommendation performances. It is defined with the following equalization:

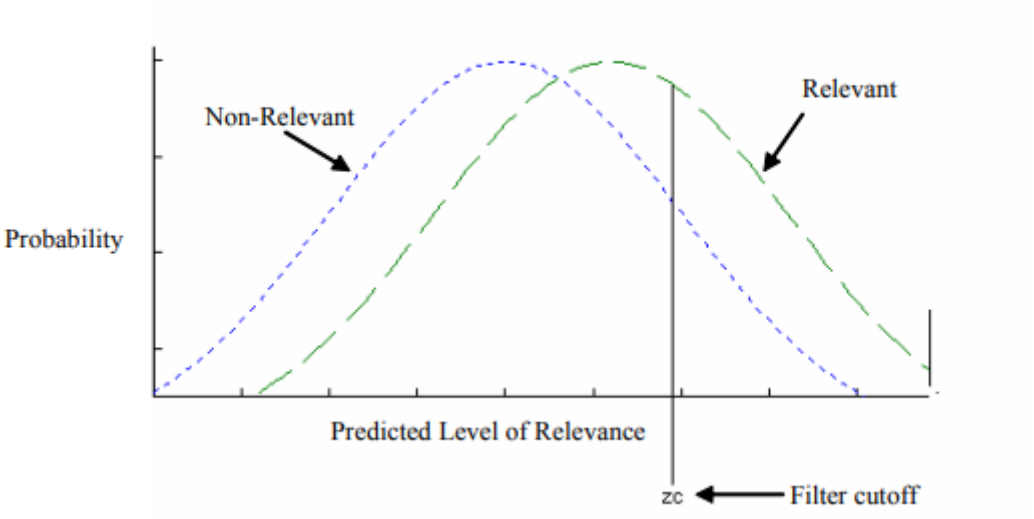
$$RMSE = \sqrt{\frac{1}{n} \sum_{\{i,j\}} \left(p_{i,j} - r_{i,j} \right)^2}$$

Where n is the total number of ratings from all users, $p_{i,j}$ is the predicted rating for user *i* on item *j* and $r_{i,j}$ is the actual rating. This metric amplifies the contributions of the absolute errors between the predictions and the true values.

ROC Sensitivity

ROC stands for Receiver Operating Characteristic. The ROC mode tries to measure the scale an information filtering system can successfully be differentiated between signal and noise.

It is presumed that a predicted level of relevance will be assigned to every potential item by the information system. On the diagram that follows, it portrays two distributions. The distribution on the left represents the probability that the system will predict a given level of relevance for an item that is not related to the information needed (X-axis). The other distribution on the right, indicates the same probability, but for relevant items.



8: Receiver Operating Characteristic Diagram

ROC sensitivity is a measure of the diagnostic power of a collaborative filtering system. Operationally, it is given by the *Area Under the ROC Curve (AUC)*. To estimate the AUC, we can use the following equalization:

$$AUC = \frac{S_0 - n_0(n_0 + 1)/2}{n_0 n_1}$$

Where n_0 and n_1 are the numbers of negative and positive examples, and $S_0 = \Sigma_{ri}$, where r_i is the rank of ith positive example in the ranked list. Given the above equation, the AUC is a measure of the quality of ranking. About the performance of the recommender system. Is important to note that the bigger AUC value is the better.

WHERE2NEXT – Application

To study the recommendation techniques, an android application will be developed. With this, the logged in user will be able to search any store, tourist attraction or any other accommodation of his choice. Also, after the first login, the application will provide more accurate and desired results to the user, to make this more personalized experience. For this purpose, collaborative filtering will be used.

Foursquare API

The Foursquare API, is a RESTful API that is based on HTTP requests with JSON responses.

Combined with Places API, it provides us with all the information about any shop or place we select. From the location of the shop to how many customers are hosted at the current time. Comments and ratings are available, also pictures uploaded from the business or the users.

Application - Step by Step

In order to make a call to the Foursquare API and get the best results for the user we need to collect some information about the user's profile and taste. When the user logs in the applications, he or she has to fill the registration form with the required information such as *fullname, username, date of birth, genre, nationality etc*. On the next step, it is requested to answer some questions in order to complete its profile information. These questions will help us understand the user's taste for the *price range* and the *distance* that he or she requires.

When this procedure is completed, we will have collected all the basic information for the user and have it stored in the database. After that, the user will have access to the category menu, in order to do its research. In the next step, the application will send the selected information, which includes the *location* of the user, via the device's GPS and the selected *category*, to the backend project, in order to retrieve the results.

Backend Project

In this project, the result calculation will be developed. Firstly, a Restful API will be developed for the successful communication to be achieved between the android application and the backend project. This API will be used to get the information from the selection that was made by the user and to make the required calls to Foursquare API in order to get the results. After we retrieve the information, a hybrid collaborative filtering will be used in order to filter them and get the best matches for the user. These results will be returned to the android application and will be displayed on the result page of the application.

Foursquare API Endpoints

There are six API Endpoints.

- Place Search
- Place Photos
- Place Details
- Place Tips
- Nearby Places
- Autocomplete

Place Search

With this call, we are going to retrieve results based on the user-submitted keywords. The matches are against names, categories, tips, tastes and sorting options that support nearby searches.

Place Photos

This endpoint will return photos that are related to the specific place. This will offer the flexibility to display a gallery or a single best photo.

Place Details

To retrieve all the information for the specified Place, including hours and category taxonomy with over 400 categories, the Place Details endpoint will be used.

Place Tips

This endpoint returns all the detailed tips for the selected Place, generated by consumer application users.

Nearby Places

Nearby Places endpoint matches users to the most appropriate POIs, based on their location for check-ins, geo-tagging photos, etc. Location matching is only available in API format from Foursquare.

Autocomplete

This endpoint returns a list of top places, addresses and searches partially matching the provided keyword and location inputs.

Which endpoint will be used in order to retrieve the list of places, will be based on the received information. If the user has selected a category, the Place Search Endpoint will be used, else if the category information is null, the Nearby Places endpoint is the next option.

For the Place Search we are able to pass the below parameters, given the user's information.

- ***LI:*** The ll as in latitude and longitude. This information will be sent from the android device, as it will be the user's current location.

- ***Radius***: This defines the distance, in meters, within which to bias place results. The maximum allowed value is up to 100.000 meters. This will be the distance that is saved under the user's profile in the database.

- ***categories:*** If the category information is not null, the categories parameter will be added. This way, we will get results related to the passed category.

- ***Min_price & max_price:*** The results will be restricted to only the places within the specified price range. The value range is between 1 to 4.

- ***open_now :*** Only the places that are open that time will be returned. The places that do not have this information, will not be included in returned values.

- ***Limit:*** Limits the number of results that are to be returned. The default value is 10 and the maximum is up to 50.

- ***Session_key:*** A user-generated token to identify the session. Basically is used for billing purposes.

For the Nearby Places, location (***ll***) and ***limit*** will be passed, in order for the places that are around the given user's location.

The ***Place Details, Place Photos and Place Tips*** will be called, once the user selects the place from the returned filtered list of places. For all the above, the **fsq_id** is required in order to make the API call. The ***session_key*** parameter will also be passed.

Filtering Algorithm

In order to provide the best result for the user, the results will be filtered with the help of a hybrid collaborative filtering based on the personality of the user. The *Personality Diagnosis Algorithm*. With this approach, we will be able to address the *cold start* problem and provide the best recommendations based on the user's profile information.

Conclusions

In this survey, some of the most known recommender techniques were analyzed. The most successful recommender techniques are the collaborative filtering algorithms. There are the memory based techniques, the model based techniques and the hybrid collaborative filtering techniques.

The memory-based techniques compute similarity between the users or items and use the weighted sum of the ratings in order to make predictions based on the similarity values. These algorithms have an easy implementation and good performance in case of dense datasets. The most commonly used algorithms are Pearson correlation and vector cosine similarity.

For the model-based technique, is important to train the algorithm model with data in order to make accurate predictions for the collaborative filtering tasks. With this technique the action of the user to take or not the recommendation is incorporated to the model, and the solution is for the function of the rewarded stream to be maximized. The most common problem of this technique is the sparsity of the data.

Finally, the hybrid collaborative filtering techniques, combine collaborative filtering methods with content-based techniques or other system recommenders in order for the predictions and the recommendation performance to be improved. For this case, they use external context information that is not available. Personality Diagnosis is a collaborative filtering approach that combines model and memory based techniques in order to produce the most accurate predictions based on the personality type of the user.

As for future improvements, the collaborative filtering algorithms, must be able to overcome, challenges like shilling attacks and noisy data and produce accurate predictions. With the rapid development of technology, is important to also improve their performance, in order to be effectively applied on any application.

Bibliography

1. M. Deshpande and G. Karypis, “Item-based top-N recommendation algorithms,” ACM Transactions on Information Systems, vol. 22, no. 1, pp. 143–177, 2004.
2. A. Nakamura and N. Abe, “Collaborative filtering using weighted majority prediction algorithms,” in Proceedings of the 15th International Conference on Machine Learning (ICML ’98), 1998.
3. J. Breese, D. Heckerman, and C. Kadie, “Empirical analysis of predictive algorithms for collaborative filtering,” in Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI ’98), 1998.
4. K. Miyahara and M. J. Pazzani, “Improvement of collaborative filtering with the simple Bayesian classifier,” Information Processing Society of Japan, vol. 43, no. 11, 2002.
5. B. Shen, X. Su, R. Greiner, P. Musilek, and C. Cheng, “Discriminative parameter learning of general Bayesian network classifiers,” in Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, pp. 296–305, Sacramento, Calif, USA, November 2003.
6. R. Greinemr, X. Su, B. Shen, and W. Zhou, “Structural extension to logistic regression: discriminative parameter learning of belief net classifiers,” Machine Learning, vol. 59, no. 3, pp. 297–322, 2005.
7. Su, Xiaoyuan and Khoshgortaar, Taghi. M, "A Survey of Collaborative Filtering Techniques"
8. JL Herlocker, "Evaluating Collaborative Filtering Recommender Systems"
9. Foursquare Developer’s Documentation
10. Wikipedia
11. David M. Pennock,Eric Horvitz ,"Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach"
12. Nagesh Singh Chauhan,"An introduction to the DBSCAN algorithm and its implementation in Python."
13. Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, Jörg Sander,"OPTICS: Ordering Points To Identify the Clustering Structure"
14. Wp.nyu.edu , “The NYU Dispatch”
15. “JavaPoint”
16. “Coursera”
17. “Real Python”