International Conference On Emanations in Modern Technology and Engineering (ICEMTE-2017) ISSN: 2321-8169
Volume: 5 Issue: 3 62 - 65

Recommender System in Machine Learning

Nishigandha Karbhari
Department of Computer Engineering
Shree L.R. Tiwari college of engineering
Mira Road, Mumbai, India
nishi_karbhari@yahoo.com

Prof. Asmita Deshmukh
Department of Computer Engineering
K.C.College of Engineering
Thane, India.
asmitadeshmukh7@gmail.com

Prof. Vinayak D. Shinde
Department of Computer Engineering
Shree L.R. Tiwari college of engineering
Mira Road, Mumbai, India
vdshinde@gmail.com

Abstract **-** Recommendation is a particular form of information filtering that exploits past behaviors and user similarities to generate a list of information items that is personally tailored to an end-user's preferences. As recommendation system brings together the main international research groups working on recommender systems, along with many of the world's leading e-commerce companies, it has become the most important annual conference for the presentation and discussion of recommender systems research. Recommender system reduce information overloading by estimating relevance.

Index Terms - *Machine learning, recommender system, content filtering.*

*****

I. Introduction

Learning is the process of knowledge acquisition. Humans learn from the experience because of their ability to reason. In contrast, computers do not learn by reasoning, but can learn with algorithms. Today, there are a large number of ML algorithms proposed in the literature. They can be classified based on the approach used for the learning process. The four main classifications are supervised, unsupervised, semi-supervised, and reinforcement learning.

Supervised learning algorithms are provided with the help of training data and correct answers. The task of the ML algorithm is to learn based on the training data, apply knowledge that was gained in the real data. Example consider an ML learning algorithm used in book classification in bookstore. A training set can be a table relating information about each book. The information about each book may be title, author, or even every word a book contains. The ML algorithm learns with the training set. When the new book arrives at the bookstore, the algorithm can classify it based on the knowledge about book classification it includes.

Unsupervised learning, ML algorithms do not include a training set. They are presented with data about real world and learn from that data on their own. Unsupervised learning algorithms mostly focus on finding all hidden patterns in data. Suppose ML algorithm access to user profile information in a social network by using an unsupervised learning approach. The algorithm can separate users into personality categories, suppose outgoing and/or reserved, allowing the social network company to target the advertising more specifically to the intended groups of users.

ML algorithms can also be classified as semi-supervised algorithm. Semi-supervised learning occurs when it work with a training set with finding missing information, and still learn from it. Example is when an ML algorithm is provided with the movie ratings. Not every user rates all movie and so there are some missing information. Semi supervised learning algorithms can learn and draw conclusions which is with the incomplete set data.

ML algorithms have a reinforcement learning approach. Reinforcement learning happens when algorithms learn based on external feedback given by a thinking entity. This approach is analogous to teaching dogs to sit or jump. When the dog performs the action correctly, the dog receives a small cookie which is positive feedback. It does not receive any cookie its negative feedback if it performs the wrong action. As an example in the computer science field, consider an ML algorithm that plays games against opponent. Moves that lead to victories are positive feedback in the game should be learned also repeated, whereas moves that lead to losses are negative feedback and should be avoided.

ML has become popular with is recent increase in processor speed and memory size. As a result, the field has a large number of algorithms now that can be used for mathematical or statistical analysis to learn, draw conclusions. The number continues to increase and thus provides an evidenced by number of scientific publications proposed for ML algorithms. For that reason, ML algorithms should have categorized based on purpose for which they have been designed. Some examples of classification can be found in although the field still does not have any standards. It's still improvising on it.

TABLE I

SR NO	Recommender System	
	Model Based Algorithm	Memory Based Algorithm
1	Probability problem: Personality Diagnosis. Enhancement to memory-based algorithms: Netflix Prize Data Linear algebra problem: Linear Algebra Operation	Pearson correlation coefficient: Similarity Measurements Predicting Ratings : Rate a Movie

2	Advantages : Scalability Prediction speed Avoidance of overfitting	Advantages : Quality Simple Implementation Easy to update
3	Disadvantages : Inflexibility Quality of predictions not accurate	Disadvantages : Slow Prediction Large memory requirement Passive output Do not generalize data

II. THE SURVEY OF RECOMMENDER SYSTEM

Recommender systems or RS use artificial intelligence or AI methods to provide users with item recommendations. Example, online bookshop use machine learning or ML algorithm to classify books and then recommend similar books to a user buying a specific category book. RSs was introduced in 1992 when Tapestry, the first RS, appeared. It was referred with the term 'collaborative filtering' for recommendation activity. This term is still used to classify RSs. Recommender systems are divided into three main categories: collaborative, content-based, and hybrid filtering which is done depending on the information used to drive the recommendations [1] RSs using a collaborative approach considers user data while processing information for recommendation. In the above mentioned book store example, by accessing user profiles in an online music store, the RS has access to all the user data, such as the age, city, country, and songs purchased. With this, the system can identify similar music preference and suggest songs bought by similar users.

RSs with a content-based filtering approach base their recommendations on the item data they can access. For example, consider a user looking for a new computer from an online store. When a user browses computer form particular category, the RS collects information about the computer and searchers in database for computers that have similar attributes, such as hardware features, price, CPU speed, and memory capacity. The result of this search is returned to the user as recommendations. The third classification describes RSs a combination of two previous classifications into a hybrid filtering approach, recommending items based on the user and the item data. For example, on a social media network, an RS may recommend profiles with similarities to the user (collaborative filtering), by comparing their interests. Second step consist of the system which considers recommended profiles as items and thus access their data to search for new similar profiles (content-based filtering). [5] So, both sets of profiles are returned as recommendations as a result. [6]

While using a collaborative or a hybrid filtering approach, RSs must gather information about the user to develop recommendations. This activity can be done either explicitly or implicitly. Explicit user data gathering happens when user is aware they are providing their information. For instance, while registering a new online service, users has to fill in a form with their name, age, gender, address, contact and email. Other forms of explicit user data gathering are where users express their preferences by rating items using a numerical value or a preference such as a Facebook. Implicit user data gathering accesses information about the user indirectly. For example, when visiting an online store, the server at the online store exchanges messages with the user's computer, and based on that, the store's RS may know the browser the user is using, as well as the user's country.

More advanced applications also monitors details like user click and keystroke logs. Besides the common recommendation process, in which users are presented with items that might be of interest, recommendations can also be provided in other ways. Trust- based recommendations take into consideration the trust relationship that users have between them. A trust relationship is a link in a social network to a friend or a following connection. Recommendations based on friends are worth more than those that do not have trust links. Context-aware recommendations are based on the context of the user. A context is a set of information about current state of the user, like time at the user location, or their activity.

The amount of context information to be processed is high, making context aware recommendations a challenging research field. Risk-aware recommendations are a subset of context-aware recommendations and take into consideration a context where critical information is available, example user vital signs. It is risk-aware because a wrong decision may threaten a user's life or cause damage. Examples can be recommending pills to be taken or stocks the user should buy or sell.

The steps involved in the theory process are:

• Identification of the possible future conditions.
• Development of a list of possible alternatives.
• Determination of the payoff associated with each alternative.
• Determination of the likelihood.
• Evaluation of the alternatives according to some criterion then select best alternative.

Types of Learning while implementing Recommender System

There are many ways in which types of learning can be categorized in ML. In this section we will discuss the categorization of the types of learning depending on the extent of feedback. There are basically four types of learning, supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning.

III. TYPES OF LEARNING

3.1 Supervised Learning

In this type of learning the training data includes the desired outputs [1]. Supervised learning follows these steps:

• Data collection – The first step in supervised learning is the collection of training data for which we know the correct outcome provided by a teacher or oracle. For example: images for which we know the object category.
• Representation – The next step is the choice of how to represent the data. Format and values of each feature.
• Modeling - The next step is the choice of a hypothesis class a set of possible explanations for the connection between examples and classes. This is our model of the problem and this type of approach is specific to supervised learning.
• Estimation – Now we need to find best hypothesis we can in the chosen class.

• Model Selection - We may also reconsider the class of hypotheses given the outcome.

Remember that each of these steps can make or break the learning outcome.

3.2 Unsupervised Learning

In this type of learning the training data does not include the desired outputs [1]. There are many variations of the unsupervised learning method based on what function or problem the unsupervised learning approach is trying to solve.

1. Clustering – This is the task of grouping a set of objects based on their features in such a way that objects in the same group (called a cluster) exhibit more feature similarity than objects in other groups (clusters).
2. Probability distribution estimation - This is the task of trying to find the underlying probability distribution from which the samples are drawn.
3. Finding association (in features) - Association rule learning is a method designed to discover interesting relations between features.
4. Dimensionality reduction is the mapping of data to a lower dimensional space such that uninformative variance in the data is discarded. In other words a data dimensionality reduction produces a compact low dimensional encoding of a given high-dimensional data set. Clustering thus finds a subspace representing the data. Methods of dimensionality reduction include Factor Analysis and Principal Component Analysis (PCA). Associated with this task is Independent Component Analysis or Dictionary Learning which finds (small) sets of factors for observation.
5. Density estimation in statistics is the construction of an estimate, based on observed data, of an unobservable underlying probability density function. The unobservable density function is assumed to be the density according to which a large population is distributed and the data can be assumed to be a random sample from that population. Here, the typical methods include Gaussian mixture model (GMM) and graphical models.
6. Sequence analysis tries to find a latent causal sequence for observations. In bioinformatics, sequence analysis can be used to assign function to genes and proteins by the study of the similarities between the compared sequences. Sequence matching also called sequence alignment and sequence segmentation are important subtasks of sequence analysis. Important techniques used include Hidden Markov Model (for Discrete state) and Kalman filter (for Continuous state)
7. Novelty detection is the identification of new or unknown data that a machine learning system has not been trained with and was not previously aware of with the help of either statistical or machine learning based approaches. Novelty detection is concerned with recognizing inputs that differ in some way from those that are usually seen. It is a useful technique in cases where an important class of data is underrepresented in the training set.

3.3 Semi-supervised Learning

In this type of learning training data includes only some of the desired outputs [1]. Semi-supervised learning must assume some structure to the underlying distribution of data. Semi-supervised learning algorithms make use of following assumptions:

1. Smoothness assumption – If two points x1, x2 are close, then so should be the corresponding outputs y1, y2.
2. Cluster assumption – If points are in the same cluster, they are likely to be of the same class.
3. Manifold assumption - The data lie on a low-dimensional manifold.

Now that we have seen how semi-supervised learning techniques work, we now describe some common methods. These include:

• Generative Model: This is a classification with additional information on the marginal density or otherwise clustering with additional information.

• Low density separation: This Implements the low-density separation assumption by maximizing the decision boundary of unlabeled points.

• Graph based methods: This builds on the concept of manifold assumption.

• Change of Representation: This method first performs an unsupervised step on all data ignoring labels. Then we ignore the unlabeled data and perform plain supervised learning using the learnt function.

3.4 Reinforcement Learning

In this type of learning rewards are received as a result of sequential actions [1]. Supervised (inductive) learning where labeled data is available is the simplest type of learning. But how can an agent learn behaviors when it doesn't have a teacher (or labelled samples) to tell it how to perform? The agent has a task to perform and takes action but at some later point, it gets feedback telling it how well it did on performing the task. The agent performs the same task over and over again. This type of learning is called reinforcement learning. There is positive reinforcement for tasks done well while there is negative reinforcement for tasks done poorly.

Examples of such learning can be best explained with game playing where when playing chess reward comes at end of game but for games such as Ping-Pong there is a reward on each point scored. In essence it is a trial-and-error learning paradigm with rewards and punishments. In other words, learn about the system from its behavior and control it with minimal feedback. This paradigm has been inspired by behavioral psychology. The goal is to get the agent to act in the world so as to maximize its rewards. In this context the agent has to figure out what it did that made it get the reward/punishment known as the credit assignment problem.

Reinforcement learning can be used to train computers to do many tasks such as playing backgammon and chess playing or controlling robot limbs. By simulating rewards from sequence of actions, this method can be used in decision making (robot, chess machine). The method can learn action so as to maximize payoff. This is used when there is not much information in a payoff and the payoff is often delayed. We can learn from reinforcement or (occasional) rewards. This is the most general form of learning. An example is an agent learns how to play Backgammon by playing against itself; it gets a reward at the end of each game. We only get feedback in the form of how well we are doing (not what we should do).

Therefore there is no supervised output but delayed reward. Examples where reinforcement learning can be used include a robot cleaning the room and recharging its battery, robot-soccer, how to invest in shares, modeling the economy through rational agents, learning how to fly a helicopter and scheduling planes to their destinations.
The basic reinforcement learning model consists of:
•A set of environment states.
•A set of actions.
•Rules of transitioning between states.
•Rules that determine the scalar immediate reward of a transition.
•Rules that describe what the agent observes.

IV. Regularization

In machine learning and statistics regularization is used for model selection. This concept introduces additional information to prevent over-fitting or in some cases to solve ill-posed problems. Regularization penalizes models with extreme parameters, e.g., places restrictions for smoothness and bounds on the vector space form. Theoretically attempts to impose Occam's razor on the solution by simplifying the model wherever feasible. From a Bayesian point of view, regularization corresponds to imposition of prior distributions on model parameters. The simplest form of regularization is the least-squares method. Now let us discuss different techniques used for regularization.

Regularization by determining number of hidden units

In general the number of input and output units in a neural network is determined by the dimensionality of the data set. However the number of hidden units M is a free parameter and controls the number of parameters that is the weights and biases in the network. Regularization is possible where the number of hidden units M can be adjusted to achieve the best predictive performance. One possible approach to adjust this parameter is to get the maximum likelihood estimate of M that gives the best generalization performance.

Regularization using Simple Weight Decay

The generalization error is not a simple function of M due to presence of local minima. There is a need to control the network complexity so as to avoid overfitting and achieve better generalization. One method is to have a relatively large number of hidden units (M) and restrict the complexity by the addition of a regularization term. The simplest regularize can be achieved by using the concept of weight decay $\tilde{E}(w) = E(w) + \lambda/2\ wTw$. The model complexity is determined by the choice of the regularization coefficient λ.

Regularization using Early Stopping

The simple weight decay affects the scaling nature of the network. Another alternate method is to stop the process early when the error reaches an acceptable value determined by a validation set. In general the error decreases at first, and then increases as the network starts to over-fit. The training process can be stopped at the point of smallest error with validation data.

V. Conclusion

Recommender systems of various types like content-based, collaborative and hybrid methods have been developed. In this paper, we reviewed various recommender methods of the current recommendation system with their capabilities. We hope that the explanation presented in this paper will advance the discussion in the knowledge of recommender systems.

References

[1] Adomavicius, G., Tuzhilin and A., "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 734 - 749, 2005.
[2] C. Hong, T. Yi, L. Luoquing, Yuan Yuan, L. Xuelong and T. Yuanyan, "Error Analysis of Stochastic Gradient Descent Ranking," IEEE Transactions on Cybernetics, vol. 43, no. 3, 2013.
[3] M. H. Henrik, "Two notes on machine "Learning"," vol. 2, pp. 364-379.
[4] "Software engineering: The trend." vol. 28, no. 2, p. 129–137, 2004.
[5] Adomavicius, G. and A. Tuzhilin, "Context-aware recommender systems." Springer US, pp. 217-253, 2011.
[6] N. Jeff, H. G. Mehmet and J. L. Sushil, "Friends Recommendations in Social Network using Genetic Algorithm and Network.," IEEE, no. 11, pp. 2207-2214.