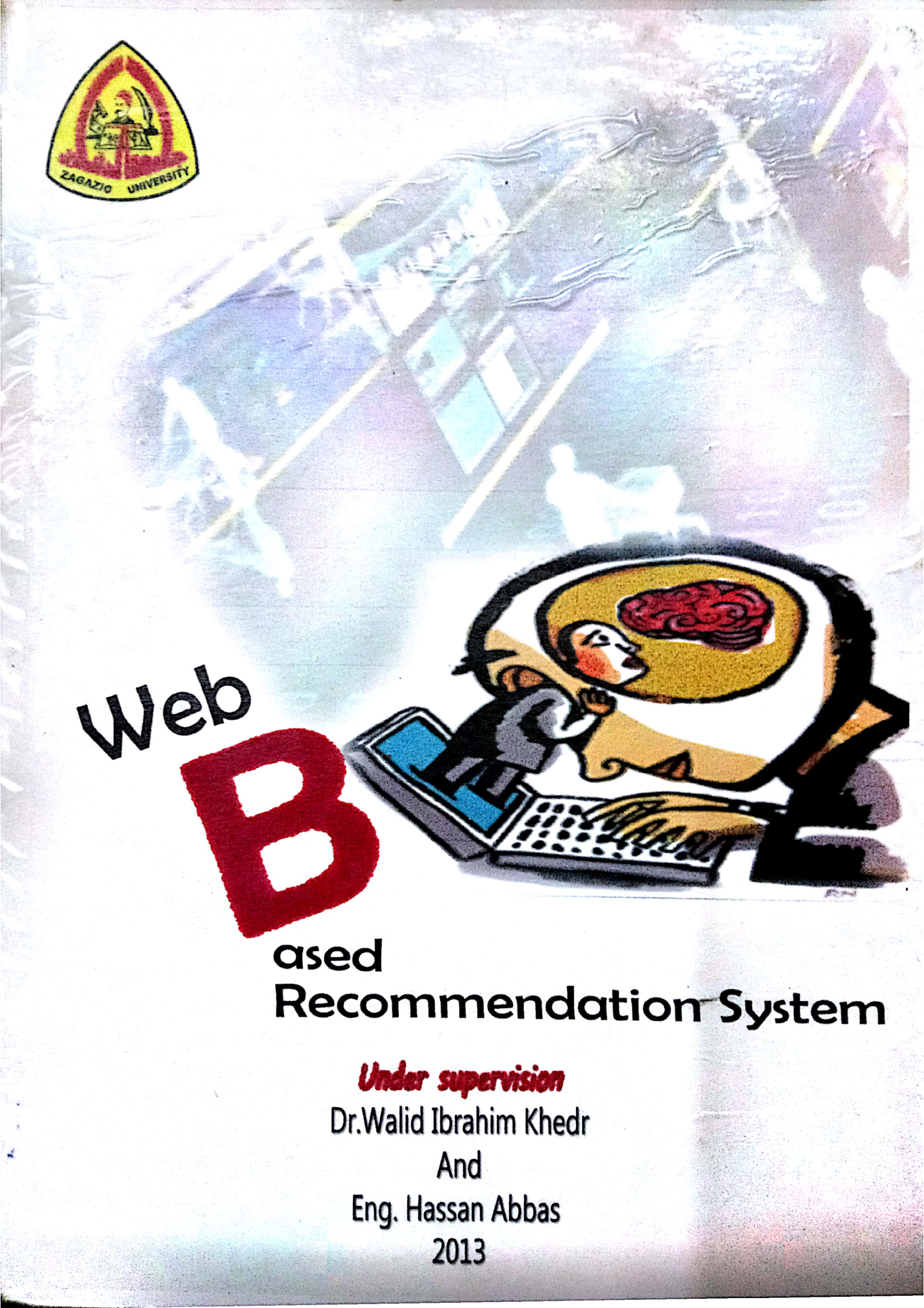


Web Based Recommendation System

Under supervision

Dr.Walid Ibrahim Khedr

And

Eng. Hassan Abbas

2013

ESHTRY

Zagazig University
Faculty of Computers and information

Web Based Recommendation System

Under supervision:

Dr.Walid Ibrahim Khedr

And

Eng. Hassan Abbas

2013

Team work:

1. **Ahmed Atef Elsayed Afifi**
2. **Ahmed Eisa Mahmoud Ali**
3. **Hassan Mustafa Abd El-latief**
4. **Mohamed Bahaa Eldin Talat**
5. **Walaa Ahmed Hussien**

Acknowledgment

Praise is to Allah "who gives us patience, brain, respect for science".
"Who taught us letters, we become slaves for him".

We would like to thank all people help us to achieve this project, even who give us just good work; We would like to thank our supervisors Dr. Walid Ibrahim Khedrand Eng. Hassan Abbas

Our deepest gratitude and sincere appreciation is for him for the suggestion of the idea, taught us how to depend on ourselves and help the others.

ESHTRY

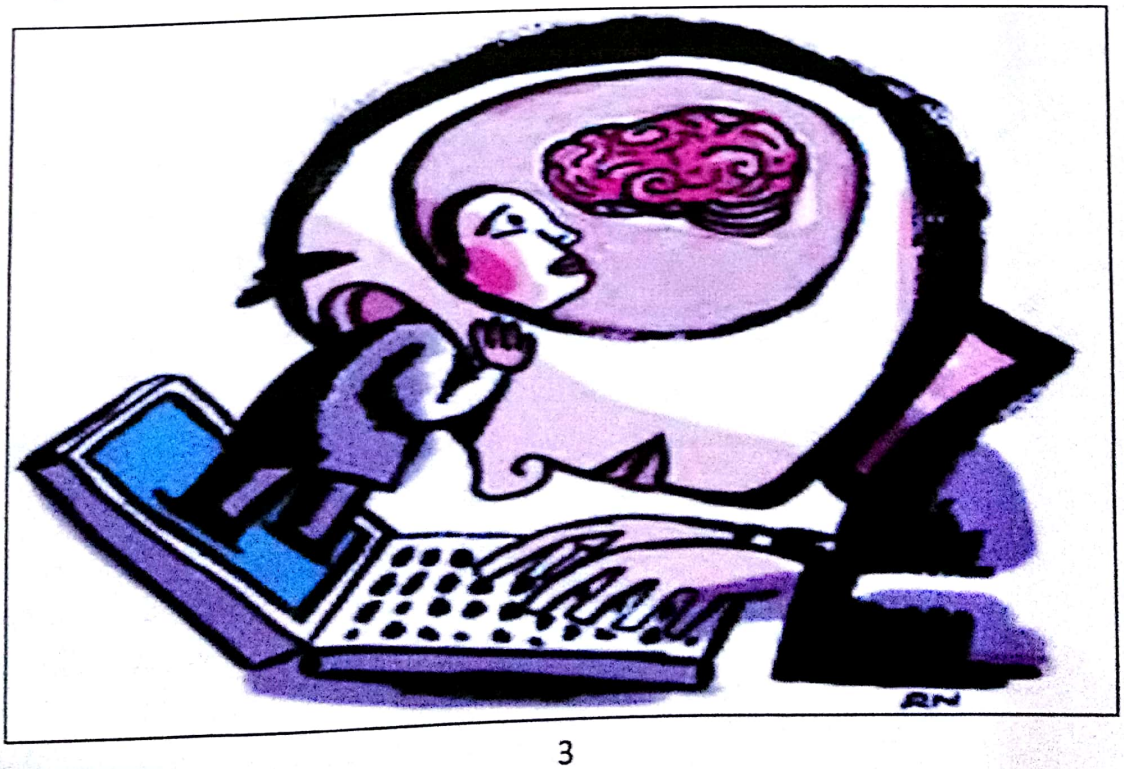
Abstract

Recommendation System

The vast amount of data available on the Internet has led to the development of recommendation systems. This project proposes the use of soft computing techniques to develop recommendation systems;Recommender systems allow online retailers to customize their sites to meet consumer tastes.

Recommender systems can now be found in many modern applications that expose the user to huge collections of items. Such systems typically provide the user with a list of recommended items they might prefer, or predict how much they might prefer each item. These systems help users to decide on appropriate items, and ease the task of finding preferred items in the collection.

Recommender systems are now popular both commercially and in the research community, where many approaches have been suggested for providing recommendations.



ESHTRY

CONTENTS AT AGLANCE

ESHTRY

CONTENTS

LIST OF FIGURES

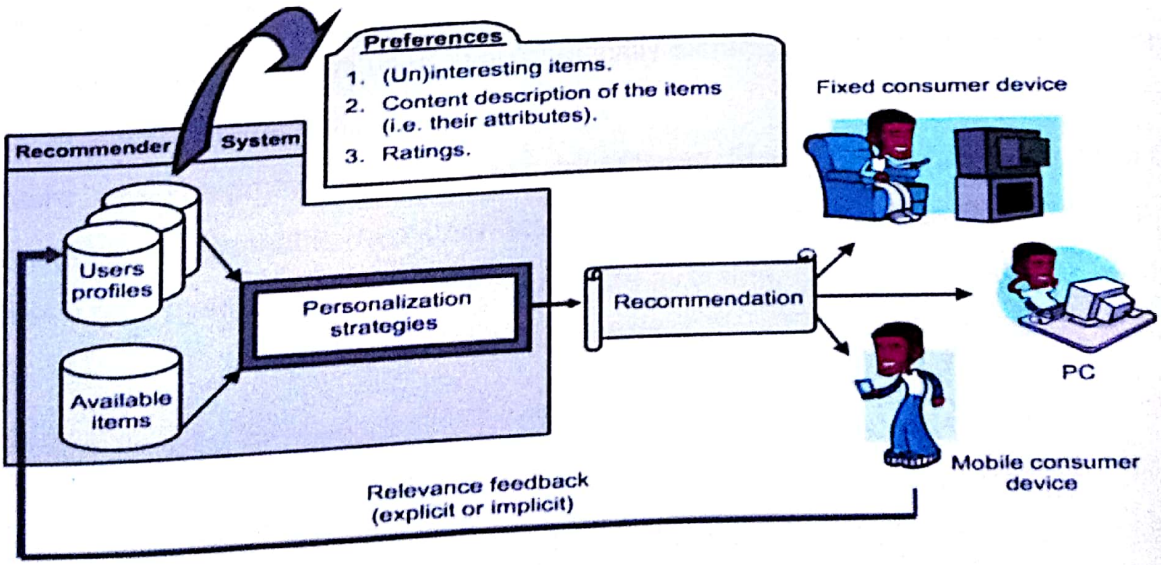
ESHTRY

Introduction

- General Introduction
- Recommender system
- Content-based recommendation
- Collaborative recommendation

The **World Wide Web** has become the primarysource of information for leisure and work activities the success of the information revolution has been largely characterized by the incredible growth in the information that is available, as the Internetand electronic media continues to expand at an incredible pace. There is adarker side to this revolution, however, as users are becoming increasinglyfrustrated by how difficult it can be to access the right information atthe right time. This is exacerbated by a number of factors beyond theobvious issues such as the sheer quantity and diversity of information thatis available.

the average Web searcher is not an information retrieval expert and cannot readily produce the sort of meaningful and informative queries that most search engines require in order to efficiently respond to a user's needs. In response to these challenges researchers have highlighted to find Recommender systems



Recommender systems, in general, aim at supporting users by recommending them previously unseen items from a (in most cases homogeneous) set of such items. Items typically encompass products, services, media items (films, music etc.), information items (as in news filtering systems), collections of information items (webpages, portals etc.).

Content-based recommendation

In general, recommender systems may serve two different purposes. On one hand, they can be used to stimulate users into doing something such as buying a specific book or watching a specific movie. On the other hand, recommender systems can also be seen as tools for dealing with *information overload*, as these systems aim to select the most interesting items from a larger set. Thus, recommender systems research is also strongly rooted in the fields of *information retrieval* and *information filtering*.

Content-based recommendation is based on the availability of (manually created or automatically extracted) item descriptions and a profile that assigns importance to these characteristics. If we think again of the bookstore example, the possible characteristics of books might include the genre, the specific topic, or the author.

In the context of content-based recommendation, the following questions must be answered:

- How can systems automatically acquire and continuously improve user profiles?
- How do we determine which items match, or are at least similar to or compatible with, a user's interests?
- What techniques can be used to automatically extract or learn the item descriptions to reduce manual annotation?

Content-based recommendation has two advantages. First, it does not require large user groups to achieve reasonable recommendation accuracy. In addition, new items can be immediately recommended once item attributes are available.

Collaborative recommendation

The basic idea of these systems is that if users shared the same interests in the past – if they viewed or bought the same books, for instance – they will also have similar tastes in the future. This technique is called *collaborative filtering*

Today, systems of this kind are in wide use and have also been extensively studied over the past fifteen years. We will cover the *collaborative filtering* techniques in the following chapter and will cover too. Typical questions that will cover in the context of collaborative approaches:

- How do we find users with similar tastes to the user for whom we need a recommendation?
- How do we measure similarity?
- What should we do with new users, for whom a buying history is not yet available?
- How do we deal with new items that nobody has bought yet?
- What if we have only a few ratings that we can exploit?

Hybrid approaches

We will discuss the question of what a "good" recommendation is laterfor instance, community knowledge exists anddetailed information about the individual items is available, a recommendersystem could be enhanced by hybridizing collaborative or social filtering withcontent-based techniques.

When combining different approaches within one recommender system, the following questions have to be answered and will be covered in the chapter on hybrid approaches:

- Which techniques can be combined, and what are the prerequisites for a given combination?
- Should proposals be calculated for two or more systems sequentially, or do other hybridization designs exist?
- How should the results of different techniques be weighted and can they be determined dynamically?

2- Recommendation System

- Recommendation system
- Content-based recommendation
- Collaborative recommendation
- Hybrid recommendation
- Neural network
- Boltzmann machines

Every time a user logins to their website, a new list of recommended items are showed based on past user's reviews or purchases. Instead of spend time navigate on the website and search for the items, a recommender system can save time for the user by display the list of items which the user likes based on user's profile. Every time a user logins to their website, a new list of recommended items are showed based on past user's reviews or purchases. Instead of spend time navigate on the website and search for the items, a recommender system can save time for the user by display the list of items which the user likes based on user's profile.

Recommender system can give personalize feeling to the user because it is based on the real input from the user and it is always update. Whenever the user buys or reviews new item, a new recommended list is created for that particular user.

There are two groups in recommender systems, *content-based* and *collaborative filtering (CF)* algorithms, and will discuss in the next section.

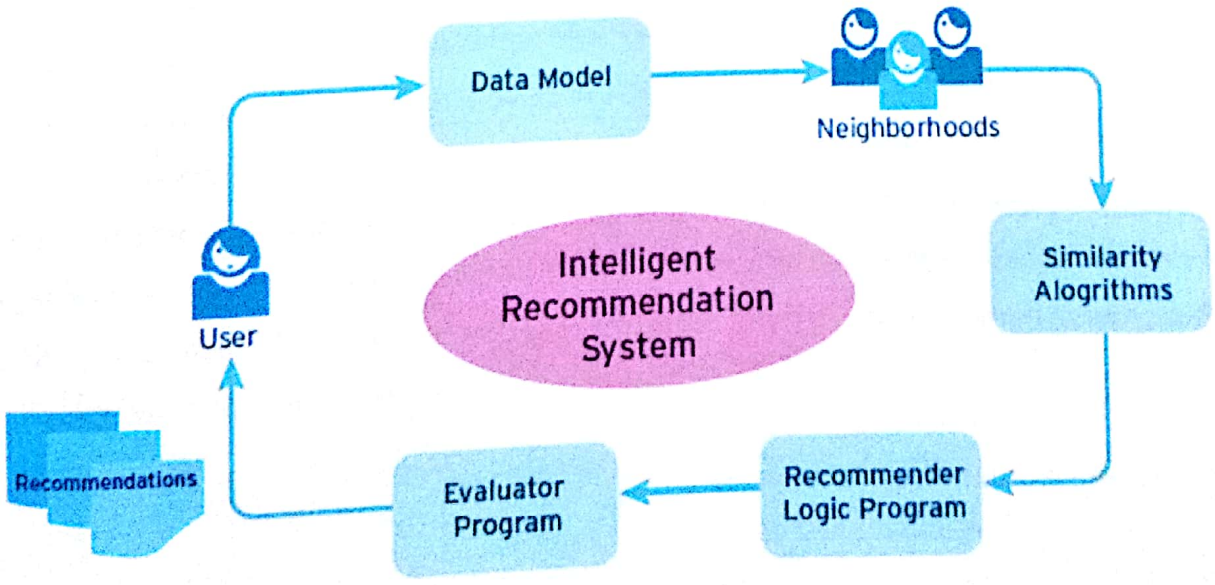


Figure1: Recommendation System Execution Flow

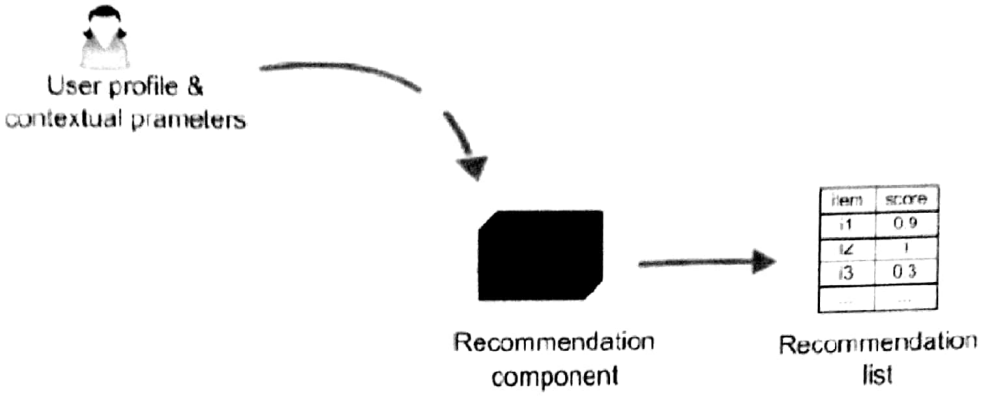


Figure2: Personalize recommendao n

Use user's profile to find matching items with the user. For a twenty three year old user, a content-based algorithm will select all items which are interested by this age. Content-based approach also can use item's profile to recommend item to user. For example, a content-based recommender system can recommend list of movies to user base on movies' genre which user's interest.

Content-based recommendation method use extra information of user's profile or item's profile in the computation. To give recommendation to one user, the profile of target user will analyze and items which matching to user's profile will be selected
For user who likes action movie, all movies with action genre will be selected and recommend to target user. In another example, when user is twelve year old and likes animation movie, then most of the Disney animation movies will be recommended to this user.
Content based algorithm can work best with items that has lots of information like documents or news website. There are some disadvantages with the content based algorithm, because its algorithm is based on user's or item's profile. The profile needs to be easy to extract by computer. Therefore, it works well with text or xml file but has difficulty when dealing with media data like movies or pictures.

2.2 Collaborative Filtering algorithm:

The main idea of collaborative recommendation approaches is to exploit information about the past behavior or the opinions of an existing user community for predicting which items the current user of the system will most probably like or be interested in.uses past user's behaviors to recommend items to user. These behaviors include user's transactions or product rating. Example, the transactions where users buy some products or the number of ratings which users review items. They don't need the explicit profiles of each user or item. For a user X who rate five on all five movies. A CF system will analyze the data and find all users who give the same five movies with rating of five then recommend the list of movies that these same users' interest to user X.

Pure collaborative approaches take a matrix of given user–item ratings as the only input and typically produce the following types of output: (a) a (numerical) prediction indicating to what degree the current user will like or dislike a certain item and (b) a list of *n* recommended items.

2.2.1 User-based Collaborative Filtering

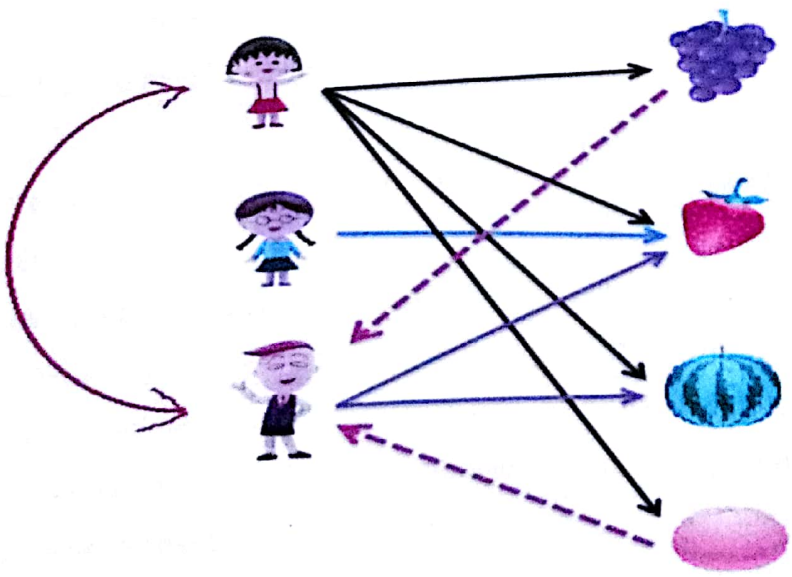


Figure 3: User-based Collaborative filtering

User-based Collaborative Filtering is one of the most chosen algorithms to use in recommender systems by online companies. It relies on the similarly behaviors between each users in the group. These behaviors are including buying or ratings items. The behaviors of various users in one group can help recommending other users in same group to buy or rate different items.

There are many algorithms to calculate the similarity between the two users in CF systems. One of them is Pearson correlation algorithm. It is a most chosen algorithm to use in CF systems. Pearson correlation only computes the similarity between the two users who rate a same item. For example, let S is the set of items where both user x and user y rated. Then the Pearson correlation computes the similarity between user x and user y as:

$$sim(x,y) = \frac{\sum_{s \in S_{xy}} (r_{x,s} - \overline{r_x})(r_{y,s} - \overline{r_y})}{\sqrt{\sum_{s \in S_{xy}} (r_{x,s} - \overline{r_x})^2 \sum_{s \in S_{xy}} (r_{y,s} - \overline{r_y})^2}}$$

User-based recommendations algorithm:

- If User A likes Items 1,2,3,4, and 5,
- And User B likes Items 1,2,3, and 4
- Then User B is quite likely to also like Item 5

Considering as the most used algorithm in Collaborating Filtering, there are some limitations in user-based approach.

- The first limitation is the scalability of the algorithm. The computation of user-base CF is more complex when the number of users gets bigger. Therefore, it is difficult to use user-based CF in big online service companies as Amazon and Netflix. User-based CF recommender systems can work very well with a small dataset, but they usually don't work well with a large dataset.

- Second limitation of user based CF is performance Its performance is slow because User-based CF needs to recomputed the similarity of user-user every time it gives new recommendation.

2.2.2 Item-based Collaborative Filtering

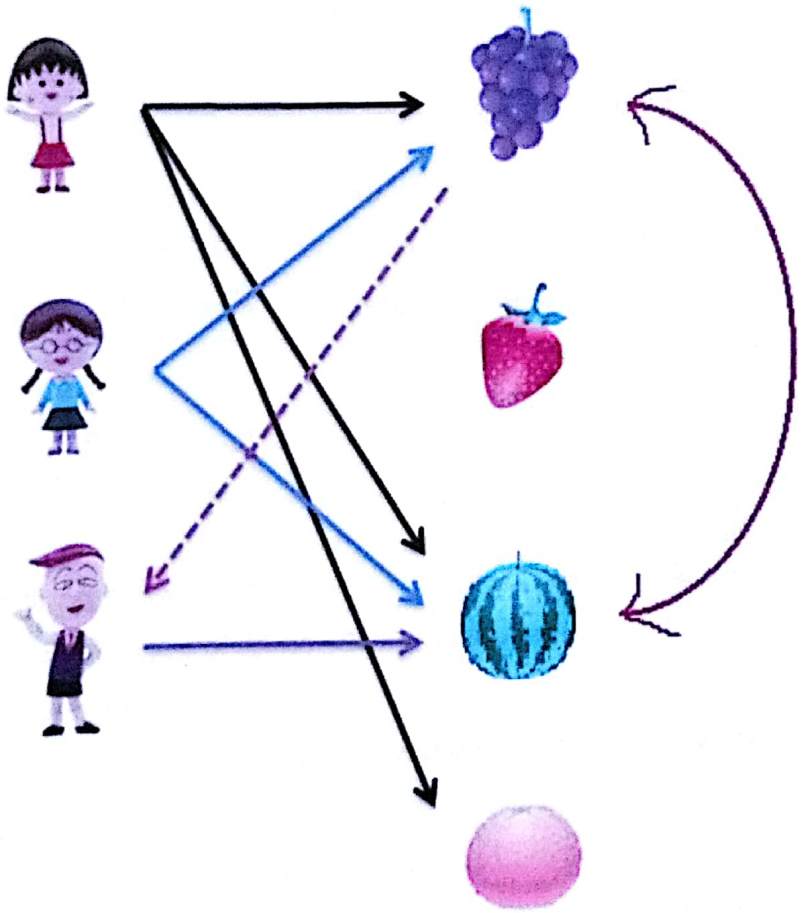


Figure 4: Item-based collaborative filtering

Instead of computation between two users, the item-based collaborative filtering algorithm computes the similarity between two items. The computation of item-based algorithm is much simpler and more scalability than user-based algorithm. Usually, there is less number of items than users in online service companies.

To compute the similarity between two items, the users who rated both items need to be selected the calculation will be used on these users and items. For Pearson correlation algorithm, the similarity of two items is compute by:

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_i)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}_j)^2}}. \quad (1)$$

Here $\bar{R}_j$is average number of item i, Ru,i is number of rating user u gives on item i.The prediction of user on target item is computed after we have similarity score of all other items to target item. For the set of all items which rated by the user, the prediction of user u on item i is given by:

$$P_{u,i} = \frac{\sum_{\text{allsimilar items, N}} (S_{i,N} \times R_{u,N})}{\sum_{\text{allsimilar items, N}} (|S_{i,N}|)}$$

Where $S_{i,n}$ is the similarity between item i and other item in set N. $R_{u,N}$ is the rating of user u on item in set N. Set N is the set of items which rated by user u.

Item-based recommendations:

- If Users who purchase item 1 are also disproportionately likely to purchase item 2
- And User A purchased item 1
- Then User A will probably be interested in item 2

Collaborative filteringapproaches often suffer from three problems: cold start, scalability, and scarsity.

- **Cold Start:** These systems often require a large amount of existing data on a user in order to make accurate recommendations.
- **Scalability:** In many of the environments that these systems make recommendations in, there are millions of users and products. Thus, a large amount of computation power is often necessary to calculate recommendations.
- **Sparsity:** The number of items sold on major e-commerce sites is extremely large. The most active users will only have rated a small subset of the overall database. Thus, even the most popular items have very few ratings.

After we know some details about two types of recommendation system we have to compare between User-based and Item-based recommender systems:

There are two major factors that come into play when talking about how user-based recommender systems compare to item-based recommender systems - the quality of the results and the performance results.

In an experiment performed by the GroupLens Research group, it was conclusive that based on building a user-based and an item-based recommender system, the item-based recommender system provided better quality of results with a lower MAE (Mean Accuracy Error) than the user-based recommender system(Sarwar, et al. 2001).

In terms of performance, the GroupLens Research group has also concluded that an item-based recommender system also has better performance over a user-based system since the item neighborhood is fairly static and can potentially be pre-computed, which results in higher online performance(Sarwar, et al. 2001)

2.3 Hybrid Recommender System

Hybrid recommender systems combine two or more recommendation techniques to gain better performance with fewer of the drawbacks of any individual one. Combining collaborative filtering and content-based filtering could be more effective in some cases. Hybrid approaches can be implemented in several ways: by making content-based and collaborative-based predictions separately and then combining them; by adding content-based capabilities to a collaborative-based approach (and vice versa). Several studies empirically compare the performance of the hybrid with the pure collaborative and content-based methods and demonstrate that the hybrid methods can provide more accurate recommendations than pure approaches. These methods can also be used to overcome some of the common problems in recommender systems such as cold start and the sparsity problem.

2.4 Neural Networks

2.4.1 What is Neural Networks?

The neural network is an information processing technology inspired by the studies of the brain and network system. In other words a neural network is a technology that tries to mimic the knowledge accusation and organizational skills of the human brain. The neural networks consist of an array of simple processors or neurons with links or connections between them. Input data is processed through this array to get the output or result. There are many other definitions of a neural network, such as a mathematical model composed of a large number of processing elements organized into layers or a new form of computing, inspired by biological model.

The inspiration for neural networks was the recognition that complex learning systems in animal brains consisted of closely interconnected sets of neurons. Although a particular neuron may be relatively simple in structure, dense networks of interconnected neurons could perform complex learning tasks such as classification and pattern recognition. The human brain, for example, contains approximately 10^{11} neurons, each connected on average to 10,000 other neurons, making a total of 1,000,000,000,000,000 = 10^{15} synaptic connections. Artificial neural networksrepresent an attempt at a very basic level to imitate the type of nonlinear learning that occurs in the networks of neurons found in nature.

As shown in Figure 4, a real neuron uses dendrites to gather inputs from other neurons and combines the input information, generating a nonlinear response ("firing") when some threshold is reached, which it sends to other neurons using the axon. Same figure also shows an artificial neuron model used in most neural networks.

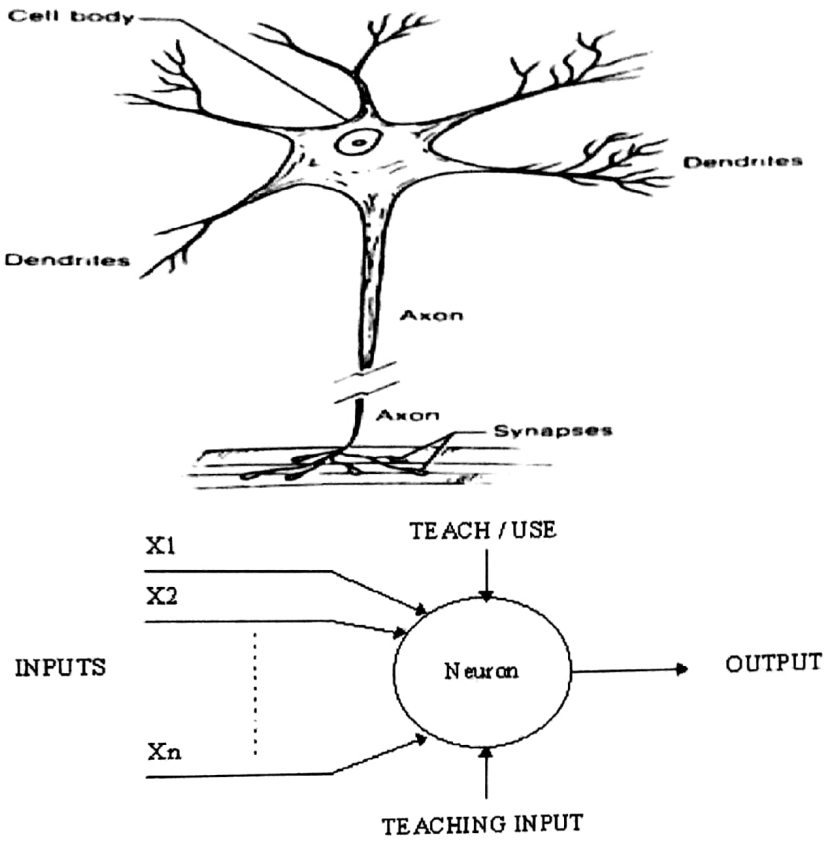


Figure 5: Human Neuron Cell and Artificial Neuron.

The inputs (x_i) are collected from upstream neurons (or the data set) and combined through a combination function such as summation ($\sum$), which is then input into a (usually nonlinear) activation function to produce an output response (y), which is then channeled downstream to other neurons.

2.4.2 Tasks of Neural Networks

Neural Networks can do several tasks of data mining like prediction, classification and clustering. All these tasks give neural networks wide use in many fields and applications. In the classification there are several problems neural networks can work with like voice recognition, handwritten recognition and face recognition. In clustering neural networks used in market segmentation, clustering gene expressions in biology and text categorization. In

prediction it used in weather forecasting, power load forecasting and stock market forecasting. Neural networks show a promising result in all these applications and another, this is because of some advantages and benefits neural networks has [6].

2.4.3 Benefits of Neural Networks

The use of neural networks offers the following useful properties and capabilities:

- Nonlinearity. An artificial neuron can be linear or nonlinear. A neural network, made up of an interconnection of nonlinear neurons, is itself nonlinear.
- Input – Output Mapping. A popular paradigm of learning called learning with a teacher or supervised learning involves modifications of the synaptic weights of a neural network by applying a set of labeled training examples or task examples. The training of the network is repeated for many examples in the set until the network reaches the steady state where there are no further significant changes in the synaptic weights. Thus the network learns from the examples by constructing an input – output mapping for the problem at hand.
- Adaptivity. Neural networks have built in capability to adapt their synaptic weights to changes in the surrounding environment. In particular, a neural network trained to operate in a specific environment can be easily retrained to deal with minor changes in the operating environmental conditions.
- Evidential Response. In the context of pattern classification, a neural network can be designed to provide information not only about which particular pattern to select, but also about the confidence in the decision made.
- Contextual Information. Knowledge is represented by the very structure and activation state of a neural network. Every neuron in the network is potentially affected by the global activity of all other neurons in the network.

- Fault Tolerance. A neural network, implemented in hardware form, has the potential to be inherently fault tolerant or capable of robust computation in the sense that its performance degrades gracefully under adverse operating conditions.
- VLSI Implementation. The massively parallel nature of neural network makes it potentially fast for the computation of certain tasks. This same feature makes a neural network well suited for implementation using VLSI technology.
- Neurobiological Analogy. The design of a neural network is motivated by analogy with the brain, which is a living proof that fault tolerant parallel processing is not only physically possible but also fast and powerful.
- Generalization. Neural networks are capable of generalization, that is, they classify an unknown pattern with other known patterns that share the same distinguishing features. This means noisy or incomplete inputs will be classified because of their similarity with pure and complete inputs.

2.4.4 Structure of Neural Networks

The following is a description of the basic components of any artificial neural network:

1- Processing Element or Neuron: An artificial neural network is composed of neurons which are the processing elements. These neurons receive inputs, process them and deliver output. The input can be raw data or the output of the other processing elements. The output can be the final product or it can be an input to another neuron. This processing element can be one of the three types:

   a. Input neuron: Each input neuron corresponds to a single attributes.

   b. Output neuron: This type of neurons is the answer to the problem that we are trying to solve.

c. Hidden Neurons: It is responsible for mapping the inputs to its corresponding outputs through the network weights.

2- Layer: It is a collection of neurons, any network consist of input and output layer and one or more than one hidden layer.

3- Weights: The key elements in any network, weights express the relative strength of the initial entering data or the various connections that transfer data from layer to layer. In other words weights express the relative importance of each input to a processing element. Weights are crucial; it is through repeated adjustments of weights that the network learns.

4- Summation function: The summation functions find the weighted average of all input elements to each processing element. $y = \sum_{i=1}^{N} x_i \, w_i$, where y is the network output, N training sample size, x_i is the i^{th} input and w_i is the i^{th} weight.

5- Transformation function: Based on the calculation received from the summation function a neuron may or may not trigger an output. The relationship between the internal activation level and the output may be linear or nonlinear. Such relationships are expressed by transformation function. There are many transformation functions and a careful selection of transformation function could determine the efficiency of the network. The most common used transformation function is the sigmoid function.

6- Learning: An artificial neural network learns from its mistakes, just like humans. The usual process of learning or training usually involves three tasks: compute outputs, compare outputs with desired answers and adjust the weights and repeat the process. The process usually starts by setting the weights randomly. The main function of learning is to minimize the difference between the actual output and desired output. To do this weights are incrementally changed. There are two types of learning:

a. Supervised Learning: In supervised learning in each input/output pair the exact difference between the desired and actual output is shown. The objective of this type of learning is to eliminate differences between actual and desired output patterns by minimizing a cost function or maximizing an objective function. In this type of learning the training process of network consist of representing input and output data to the network. The data included in the training process is called the training data set. This training phase can consume a lot of time. When the training is completed, the weights are noted and will be used in the actual operation.

b. Unsupervised Learning: Here, networks use no external influences to adjust their weights. In other words the desired output is not exist, the network looks for regularities or trend in the input signals and makes adaptations according to the function of the network.

2.4.5 Types of Neural Networks

Neural networks have several architectures (topologies) and these topologies are classified according to the following criteria:

- The direction of the processing from input to output layer into two types:
    - Feed forward which takes one direction from input to hidden to output.
    - Recurrent in which the output of some units is fed back as an input to some others.
- The connection between each layer in the network into two types:
    - Fully connected where each unit in each layer connected to all other units in the next layer.

- Partially connected where each unit can be connected to some units in the next layer not all of them.

- **Hidden layers number:**
  - Input layer, one hidden layer and output layer.
  - Input layer, more than one hidden layer and output layer.
  - Input layer and output layer only.

There are several architectures and types of neural networks appear according to the criteria we explained, each of these types is described in the following sections.

2.4.5.1Multi-Layer Perceptron

MLP network is composed of more than one layer of neurons, with some or all of the outputs of each layer connected to one or more of the inputs of another layer. The first layer is called the input layer, the last one is the output layer, and in between there may be one or more hidden layers [9]. Figure 2.2 shows the network architecture.

Properties of architecture:

1- No connections within a layer.
2- No direct connections between input and output layers.
3- Fully connected between layers.
4- Often more than 3 layers.
5- Number of output units needs not equal number of input units.
6- Number of hidden units per layer can be more or less than input or output units.

Training method:

1- Supervised learning.
2- Use back propagation algorithm as a training algorithm and other optimization techniques like GA, evolution strategy.

3- All the neurons in hidden and output layer have a nonlinear mapping by using nonlinear activation functions like sigmoid and tan activation functions.

Applications:

1- Speech recognition.
2- Face recognition.
3- Handwritten recognition.
4- Text classification.
5- Machine learning.

Advantages:

1- Generalization.
2- Fault Tolerance.

Disadvantages:

1- Computationally expensive learning process. Large number of iterations required for learning, not suitable for real time applications.
2- No guaranteed solution. During the learning process the network can stick in local minimum.
3- Scaling Problem.

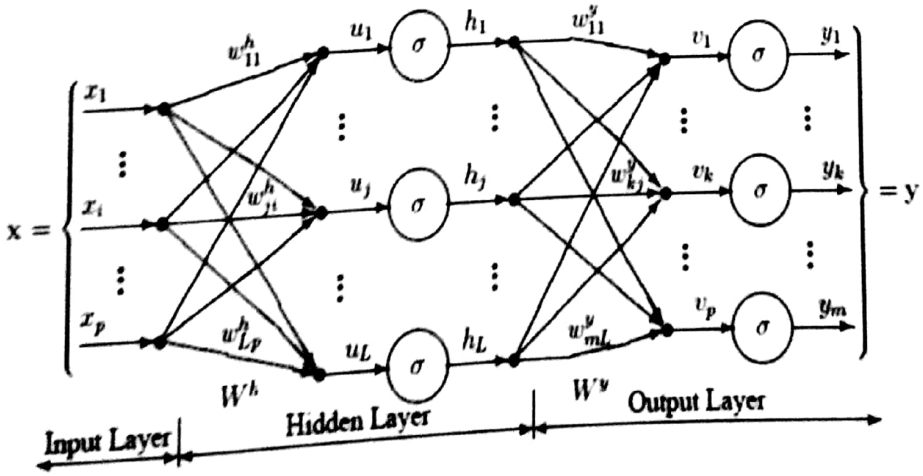


Figure 6: Multi-Layer Perceptron Structure.

2.4.5.2 Radial Basis Function Network

Compared to MLP network, the RBFN is the next-most-used network model [10]. As the name implies, this network makes use of radial functions. MLP networks and RBF networks can be used for the same types of problems, and the commands and their options are very similar.

Properties of architecture:

1- No connections within a layer.
2- No direct connections between input and output layers.
3- Fully connected between layers.
4- There are only 3 layers.
5- The input to the network is nonlinear but the output is linear.
6- The hidden layer activation functions is the radial basis functions.

Training properties:

1- Supervised learning.
2- The activation functions of the hidden units are fixed radial basis functions and these functions have centers which calculated from the input samples.
3- Each hidden unit outputs is the distance between the inputs and these centers by using a radial basis function like Gaussian.
4- The output is a linear combination of all the hidden units.
5- The error is calculated by finding the difference between the desired and actual output.
6- The weights of the network are adjusted with respect to minimize the error function using the same rules in the back propagation algorithm used with MLP.

Applications:

1- Image processing.
2- Speech recognition.

3- Time series analysis.

4- Radar point source location.

5- Medical diagnosis.

6- Pattern recognition.

Advantages:

1- Required less time in the train process than MLP.

2- Has simple architecture with simple weights, so it has a unique solution to the weights.

Disadvantages:

1- Suffers from curse of dimensionality.

2- Has less generalization than MLP for each training example

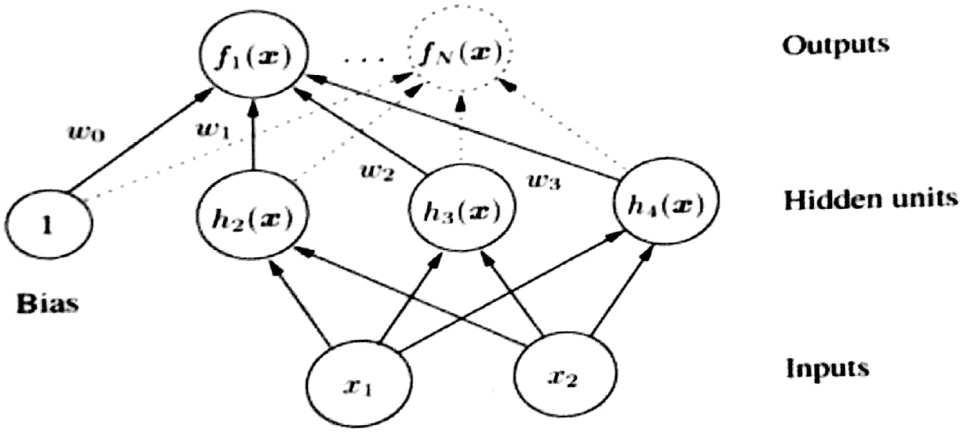


Figure 7: Radial Basis Function Network Structure.

2.4.5.3 Probabilistic Neural Networks

A type of neural network using kernel-based approximation to form an estimate of the probability density functions of classes in a classification problem [11]. In this type of network additional layer added before output layer which calculate the probability of each class. This

type of networks is used mostly in classification problems. Figure 2.4 shows the network architecture.

Properties of architecture:

1- No connections within a layer.
2- No direct connections between input and output layers.
3- One neuron in the input layer for each predictor variable.
4- One neuron for each case in the training data set.
5- Additional layer called pattern layer contains neurons represent number of classes.
6- The output layer contains only one output layer.

Training properties:

1- Supervised learning.
2- When the input is presented, the first layer computes distances from the input vector to the training input vector and produces a vector whose elements indicates how close the input is to a training input.
3- The second layer sums these contributions for each class of inputs to produce as its net output a vector of probabilities.
4- Finally, a transfer function on the output of the second layer picks the maximum of these probabilities, and produces a 1 for that class and a 0 for the other classes.
5- The error is calculated by finding the difference between the desired and actual output.
6- The weights of the network are adjusted with respect to minimize the error function.

Applications:

1- Expert systems.
2- Classification.

Advantages:

1- Much faster to train than MLP network.

2- More accurate than MLP.

3- Generate accurate predicted target probability scores.

4- Approached Bayes optimal classification.

Disadvantages:

1- Slower than MLP in classifying new cases.

2- Require more memory space to store the model.

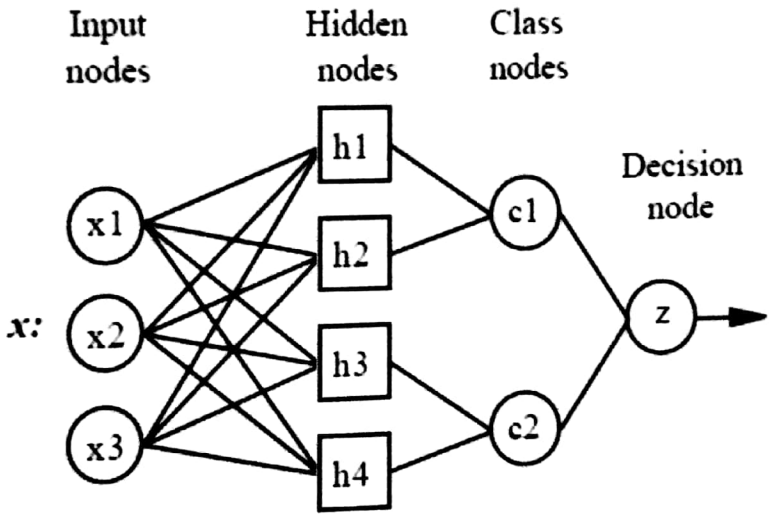


Figure 8: Probabilistic Neural Network Structure.

2.4.5.4 Recurrent Neural Networks

It is a class of neural network where connections between units form a directed cycle. This creates an internal state of the network which allows it to exhibit dynamic temporal behavior. Figure 2.5 shows the network architecture.

Properties of architecture:

1- Connections within each layer.

2- Direct connections between input and output layers.

3- Additional units in the input layer which represent the previous step of the network.

Training properties:

1- Supervised learning.

2- Back propagation algorithm is used to train this network also but in different way from MLP because the feedback connections and the additional units in the input layer.

Applications:

1- Time series prediction.

2- Pattern recognition.

3- Speech recognition.

Advantages:

1- Able to build up internal memory suited for many applications specially prediction.

2- Computationally more powerful than feed forward networks.

3- Can deal with curse of dimensionality.

Disadvantages:

1- The difficulty of developing generally applicable learning algorithm to train the network.

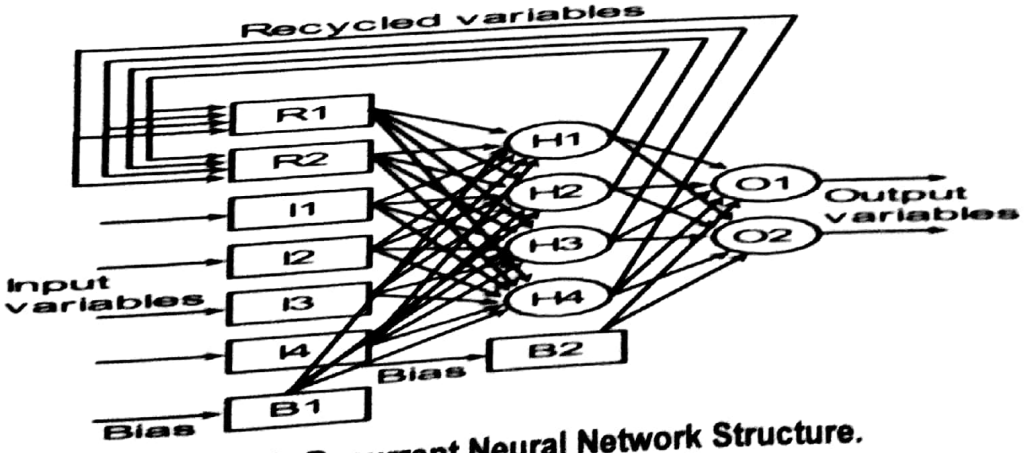


Figure 9: Recurrent Neural Network Structure.

2.4.6 Back Propagation algorithm

In back propagation algorithm we take the first and second derivative to minimize the error with respect to the parameters we want to estimate, but this algorithm can fall in local minima and is time consuming because has two phases; the forward phase which calculates the network output and the backward phase which compares the network output and the actual output, based on that difference the network weights are updated and so on until the error percentage reached a selected tolerance.

The algorithm starts with initial weights and iterate through the training set to fit these samples and find the best weights. The summary of the algorithm is showed in the following steps:

1- Present a training sample to the neural network.
2- Compare the network's output to the desired output from that sample and calculate the error in each output neuron.
3- Adjust the all the weights of the network backwardly by using the algorithm equations which adjust these weights.
4- Repeat these steps until the training set is finished or the acceptance error reached.

The back propagation algorithm is the most common used algorithm in training the most of neural networks, but it suffers from falling in local minim and we must know the first and second derivative with respect to the error to find the equations which adjust the weights of the network.

Boltzmann Machines

Boltzmann machines are Hopfield networks that use hidden units with visible units but the main difference is that in Boltzmann machines is that units are stochastic binary units not deterministic binary units as shown in figure 9.

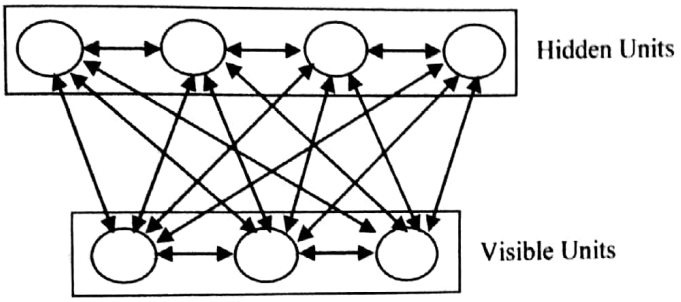


Figure 10: Boltzmann Machine Structure

The use of stochastic units helps in escaping from poor local minima. This is done by starting with a lot of noise which helps in cross the energy barriers then reduce the noise slowly to be end with global minima and this what called simulated annealing process. The following equation defines the probability of unit to be turned on:

$$P(s_i = 1) = \frac{1}{e^{\frac{\Delta E}{T}}}$$

Where ΔE the energy gap we discussed before and T is the current temperature used. Another approach to work with these models is by discarding the changing in the temperature and fix it be one. This approach depends on a concept in thermodynamics called thermal equilibrium. This concept state that in specific low and fixed temperature the system after a while reach a stable state and this state can be calculated based on the current energy of the system. This most common state of the system can be found at this fixed value of temperature. The probability of the system to be in specific state can be calculated using the current energy level in the system according to a distribution called Boltzmann distribution

and this is the reason this model is called Boltzmann Machines. In Boltzmann machines the concept of thermal equilibrium doesn't means get to the lowest energy level of the system but reach to a stable state of the system. Thermal equilibrium means reaching to stationary probability distribution. The main idea of learning this model is starting with specific distribution and by updating the states of the units stochastically we can reach to stationary distribution which remains unchanged so much. The probability of find the system in current state is calculated by the following equation: $p(v,h) = \frac{e^{-E(v,h)}}{\sum_{u,g} e^{-E(u,g)}}$

Where v is visible unit, h is a hidden units and -E(v, h) is the energy function maintained by the state of both visible and hidden units. The equation of E(v,h) is as the follows:

$$-E(v,h) = \sum_{i \in vis} v_i b_i + \sum_{k \in hid} h_k b_k + \sum_{i,j} v_i v_j w_{ij} + \sum_{i,k} v_i h_k w_{ik} + \sum_{k,l} h_k h_l w_{kl}$$

The marginal probability of observed visible state is given by the following equation:

$$p(v) = \frac{\sum_h e^{-E(v,h)}}{\sum_{u,g} e^{-E(u,g)}}$$

The dominator is called the partition function or the normalization term for the distribution. Boltzmann machines are used in modeling the data distribution. Any data is drawn from a probability distribution and by building a model that estimates this distribution can be helpful in many other applications and problems in machine learning. In this model its assumed that the observed visible data is generated or caused by another hidden units not observable. So the main purpose is to model the distribution between the observable visible data and the hidden unobservable units which responsible for capturing the regularities in the data. Boltzmann Machines can be used in several tasks like modeling the data distribution, generating the data samples from data and detecting the features form data. The learning type is unsupervised. Learning is accomplished using gradient based learning by taking the

derivatives of the marginal distribution of the observed data with respect to model parameter and this generates a learning rule for updating model weights until reaching thermal equilibrium by the following equation:

$$\frac{\partial \log p(v)}{\partial w_{ij}} = \langle s_i s_j \rangle_v - \langle s_i s_j \rangle_{model}$$

Where the first term is the expected value of product of states at thermal equilibrium when we observed the visible units. And the second term is the expected value of states at thermal equilibrium with no clamping and this term is the hardest part in this equation because it requires to get all possible configurations the system can be in. To calculate this term we need to model from the model distribution. Probability estimation algorithms can be used like MCMC methods which used to approximate the model distribution. These methods take long time to reach to an approximation for the model distribution. This problem is solved by finding a fast way to approximate this second term in updating model weights and this is what we will discuss in the next section of RBM.

Restricted Boltzmann Machines

RBM are Boltzmann machines but restricted in the connections among the visible and hidden units. The connections among the hidden units and the connections among visible units are removed. Only the symmetric connections between the visible units and hidden units are exist as shown in figure 10.

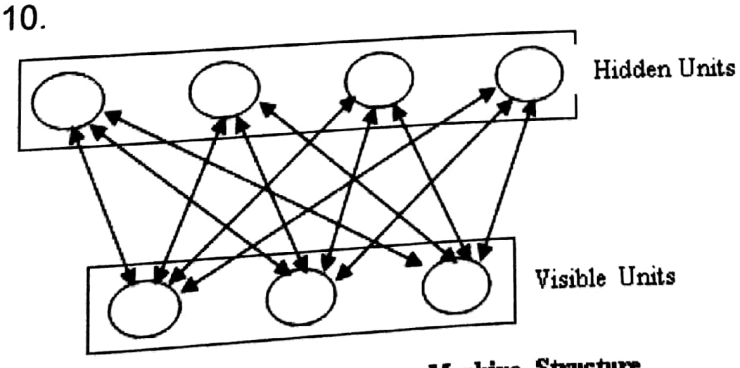


Figure 10: Restricted Boltzmann Machine Structure

The second change in the model is the learning procedure which is the same as the Boltzmann machines but the difference is in calculating the second term which takes the long time. The second term is calculated by run the Markov chain to n steps and this makes the learning of the model more fast. This shortcut gives a good and acceptable result in problems of probability distribution approximation and detecting the features from data. This learning method called Contrastive Divergence (CD). This algorithm is the main algorithm used to train RBMs. Another versions appeared to modify this algorithm and make it more faster like Persistence Contrastive Divergence (PCD) and using Fast Weights with Contrastive Divergence algorithms (FPCD). Below is pseudocode description of the CD algorithm.

Algorithm parameters:

- Learning rate (decreased linearly from initial value to zero).

- N steps of sampling

Initialization:

- Initialize model weights W with random values from normal distribution.

- Initialize bias values for both visible and hidden units.

Then repeat:

1. Get next batch of training data v^+.

2. Calculate $h+ = P(h \mid v^+, W)$.

3. Calculate positive gradient (first term in equation 7).

4. Calculate $h- = P(h \mid v^-, W))$.

5. Calculate negative gradient (second term in equation 7 for n sampling steps).

6. Calculate full gradient g = positive gradient - negative gradient.

7. Update W = W + LR * gradient.

Figure 11 shows how the gradient is calculated during learning process for 1 sampling step, in this case CD is called CD-1. If the second term is calculated with n sampling steps it is called CD-n.

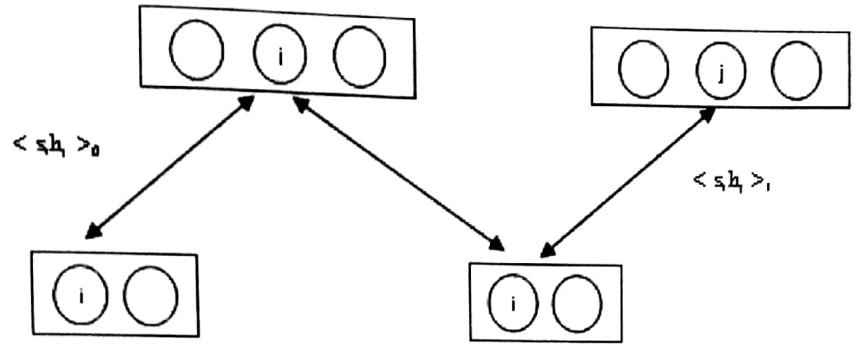


Figure 11: Contrastive Divergence used to approximate the gradient value needed to update the weights.

In PCD the chain is a persistence chain that remains with us during the training and each time the visible units used. Below is pseudocode description of the PCD algorithm.

Algorithm parameters:

- Learning rate LR (decreased linearly from initial value to zero).

- N steps of sampling

Initialization:

- Initialize model weights W with random values from normal distribution.

- Initialize bias values for both visible and hidden units.

- Initialize persistence Markov chains v^- to all zeros.

Then repeat:

1. Get next batch of training data v^+.

2. Calculate h+ = P(h | v^+, W).

3. Calculate positive gradient (first term in equation 7).

4. Calculate h- = P(h | v^-, W)).

5. Calculate negative gradient (second term in equation 7 for n sampling steps).

6. Update v- = sample from P(v | h^-, W).

7. Calculate full gradient g = positive gradient - negative gradient.

8. Update W = W + LR * gradient.

In FPCD algorithm is PCD but with additional set of weights used that cause a temporal changing in the surface of the energy function. This additional fast weights make the learning faster. Below is pseudocode description of the PCD algorithm.

Algorithm parameters:

- Learning Rate LR (decreased linearly from initial value to zero).

- Fast learning rate FLR remains constant.

Initialization:

- Initialize model weights W with random values from normal distribution.

- Initialize model fast weights FW with random values from normal distribution.

- Initialize bias values for both visible and hidden units.

- Initialize persistence Markov chains v^- to all zeros.

Then repeat:

1. Get next batch of training data v^+.

2. Calculate $h+ = P(h \mid v^+, W)$.

3. Calculate positive gradient (first term in equation 7).

4. Calculate $h- = P(h \mid v^-, W + FW)$.

5. Calculate negative gradient (second term in equation 7 for n sampling steps).

6. Update v- = sample from $P(v \mid h^-, W)$.

7. Calculate full gradient g = positive gradient - negative gradient.

8. Update W = W + LR .gradient.

9. Update WF = WF $\cdot\frac{19}{20}$ + gradient . FLR