Ontology-based Personalized Recommendation System

Student Name- Akanksha Gupta

IIIT-D MTech CSE-DE-MT12032

Indraprastha Institute of Information Technology, New Delhi

Supervisor:
Dr. Anupama Mallik

Thesis submitted in partial fulfilment of the requirements for the Degree of MTech

in Computer Science

with specialization in Data Engineering

May 2015

Certificate

This is to certify that the thesis titled **"Ontology-based Personalized Recommendation System"** submitted by **Akanksha Gupta** for the partial fulfillment of the requirements for the degree of *Master of Technology* in *Computer Science and Engineering* is a record of bonafide work carried out by her under my guidance and supervision in the Data Engineering specialization at Indraprastha Institute of Information Technology, Delhi. This work has not been submitted anywhere else for the reward of any other degree.

Dr. Anupama Mallik
Indraprastha Institute of Information Technology

Abstract

Past decade has seen a prominent rise in the number of e-commerce applications in the World Wide Web. Designing recommendation algorithms for predicting user interests is quite challenging for such systems. Several recommendation frameworks have been proposed in research. However, when it comes to recommendation of media-rich commodities, most of the algorithms designed so far, utilize the metadata associated with the digital products. Such systems may not generate correct recommendations if the metadata is insufficient or inaccurate. Our approach is motivated by the fact that by making use of a domain ontology and relating media content to domain concepts, it is possible to remove the semantic gap between high-level semantic concepts and low-level media features. This can be utilized to improve recommendation of media-rich commodities to the user, as such a recommendation is based on media content as well as metadata. In this work, we have proposed a video recommendation framework based on ontology. The multimedia ontology is represented in Multimedia Web Ontology Language (MOWL) [16], which supports a probabilistic reasoning scheme. We have also given a novel approach for personalizing the recommendations on-the-fly, by analyzing user preferences and modifying the recommendation model accordingly. We have experimented with a media-rich dataset consisting of English movie videos. Proposed system can add semi-automatic conceptual annotations to movie scenes as well as to full movies with the help of the ontology. This semantic metadata is also utilized while making recommendations to the user. The system can recommend not just full movies, but scenes from the movies based on user interest. We have illustrated the proof of concept by corroborating our system with anonymous users.The contentment score and recommendation accuracy obtained, has validated the efficiency of our approach.

Acknowledgements

First and foremost, I offer my sincerest gratitude to my supervisor, Dr. Anupama Mallik, who has supported me throughout my thesis with her patience and knowledge. I am highly obliged for the motivation and enthusiasm provided by her throughout the research and writing of this thesis. Her timely and efficient contribution helped me shape my work into its final form and I express my sincerest gratitude for her assistance in any way that I may have asked.

Working on this thesis would have been more tough without the love and support of my friends. It would be unjust not to mention a big thanks to Ankush and Lovey who have been my biggest support throughtout this journey.

Last but not the least, I'm forever indebted to my parents for their untiring faith and support in my every decision and dream.

Contents

List of Figures

List of Tables

Abbreviations

CPT	**C**onditional **P**robability **T**able
MOWL	**M**ultimedia **W**eb **O**ntology **L**anguage
OWL	**W**eb **O**ntology **L**anguage

"Dedicated to my parents...."

Chapter 1

Introduction

Due to information explosion in World WideWeb, a huge amount of digital content is available online. This makes it difficult for information systems to retrieve appropriate data by finding useful information as per the user requirements [3]. In e-commerce applications, it is quite challenging task to recommend items to a user according to his choice, especially when it comes to recommendation of media-rich commodities like movies, music etc. A lot of research has been done to predict the correct recommendations for a user. One technique is collaborative filtering [11] [22], which generates the recommendations by finding out interests of a user on the basis of preferences or taste information from similar users. Such systems try to find out similar users based on the ratings given by user, these systems become inefficient for the cases where number of users are small relative to number of items. So, there is a problem of sparse ratings. Another approach is content based filtering [21] which recommends items on the basis of metadata associated with the items. It employs techniques to represent features that are directly acquired by textual descriptions of items, and then on the basis of item-item similarities, system recommends items similar to user selection. When it comes to recommendation of media-rich commodities, most of research has been done in which system utilizes the metadata associated with items for determining the recommendations. A limited research has been done which utilizes the actual media features extracted from multimedia data in order to improve the system's efficiency. The main challenge is to incorporate low level media features in such systems so as to enhance the recommendation results.

1.1 Related Work

Majority of recent research on recommendation systems has used Collaborative Filtering technique using some association rule mining and clustering methods to find out k-Nearest Neighbors [20]. Iwahama et al [8] have proposed an algorithm for music recommendation by analyzing the feature parameters about music data in MIDI format. System performs filtering by finding out items having similar features by building a decision tree model from user's initial ratings. Decision tree categorizes a music into three categories: "like", "neutral" and "dislike". Efficiency of such system is totally limited to the low-level features extracted from media as no other related concepts have been used for recommendation. A combination of collaborative and semantic based approach has been proposed by Lecue et al [12] which performs Description Logic based reasoning on semantic descriptions of services for finding out similarity of services.However, this crisp reasoning fails when it comes to media features and doesnot provide means to deal with uncertainty in media observation.

Due to easy availability of Movielens dataset,a lot of research has been carried out in the *movie* domain. Nessel J. and Cimpa B. have proposed MovieOracle [19], which recommends movies by comparing user interests to movie dialogues gathered from movie subtitle files. Such systems work only if validated metadata is available for all movies. Deng et al [5] have proposed a hybrid recommendation methodology which combines collaborative and content-based filtering approaches based on ontology representation and neural network technology. However they consider only metadata available for movies and do not perform any video content analysis. Zhenglian et al [26] have proposed a personalized recommendation algorithm based on an ontological user interest model. An incrementally updating algorithm of user interest model is described, which is based on Spreading Activation theory. It learns current user interests and accordingly updates the user interest model. In [6], the system takes in user's personal information and predicts movie preferences. Then it performs clustering on movies and generates a long list of questions for a user to refine the recommendation. System learning is based on feedback from the user. Meshram et al [18] have suggested an approach to improve the recommendation methodology by using the audio features of movie audio features for emotion recognition and scene classification. A different ontology-based approach has been proposed by Ajmani et al [2] for apparel recommendation. It alleviates the cold-start problem but doesn't explain how the system adapts to user behavior when his choice is different from the recommended set of items.

In this paper, we have presented an ontology-based recommendation system which uses the semantics of the content as well as that of the metadata in making recommendations. Further we have personalized the recommendations based on user relevance feedback. Using MOWL [16], we have removed the semantic gap by mapping the low-level features associated with media to high level concepts in a multimedia ontology of the domain. MOWL provides a probabilistic knowledge representation and reasoning model for media-rich domain. The recommendation framework uses a probabilistic reasoning scheme for determining the user interests and makes recommendations accordingly. This proposed framework uses collaborative filtering as it considers user profile, encoded in a domain ontology, for making the initial recommendations. It personalizes the recommendations on-the-fly by monitoring the user clicks and utilizing semantic information associated with selected items. We have chosen the media-rich domain of English Movies for showcasing our recommendation framework. We have also proposed an ontology-based framework for conceptual annotation of movie videos, based on the media features associated with scenes in a movie. Since huge amount -of multimedia data available on-line has insufficient metadata associated with it, recommendations can improve if semantic information is associated with media documents. Our system handles this in a way that it first generates conceptual annotations of movie videos and then recommends items to a user based on the concepts associated with them. Our framework can recommend not just movies, but scenes or segments from the movies, based on user interest.

Rest of this paper is organized as follows: Section 2 introduces the Multimedia Web Ontology Language and details the proposed framework for Recommendation. Section 3 explains the Movie domain and knowledge encoded in the movie domain ontology. Section 4 details the Conceptual Annotation of Videos with an algorithm. Section 5 explains the algorithm for Personalized Recommendation. Section 6 outlines the experimental evaluation of proposed system. Section 7 provides some concluding remarks.

Chapter 2

Proposed Framework

2.1 Multimedia Web Ontology Language

MOWL, introduced in [16], is a new multimedia ontology representation language with a probabilistic reasoning framework. In traditional ontology representation, it is difficult to associate media content with concepts and there is no support for probabilistic reasoning. MOWL provides constructs to associate media properties with concepts in an ontology. It also provides a way to represent spatio-temporal relations between concepts and media objects in an ontology. The uncertainty associated with observation of media content can be encoded in the form of conditional probability tables (CPTs) in a MOWL ontology, which provides probabilistic reasoning with Bayesian networks.

2.2 Conceptual Annotation of Videos

Figure 2.1 shows a framework for video annotation. It associates concepts of an ontology with video segments or scenes and thus tags the videos with semantic concepts. A domain expert encodes the domain knowledge in a MOWL ontology. The training dataset comprises of video-clips of different videos manually annotated by the expert. Videos in the training set are input to the **Feature Extraction Module**, which extracts aural and visual features from a video. It also creates a video description scheme using Indicator Vectors [17]. This generated video representation for the training video clips is then input to the **Annotation Generation Module** to learn a model. Video annotations correspond to semantic concepts in the domain ontology. Using this model, machine-learning is

applied to classify test data, and generate automatic annotations. Once all the test video clips have been annotated, the annotation for full videos is done by the **Annotation Module for overall video**. It associates those semantic concepts with the full video which occur most frequently with the clips belonging to the video. These annotated videos and video clips are then considered in the movie recommendation process. sectionRecommendation of Videos Figure 2.2 shows a

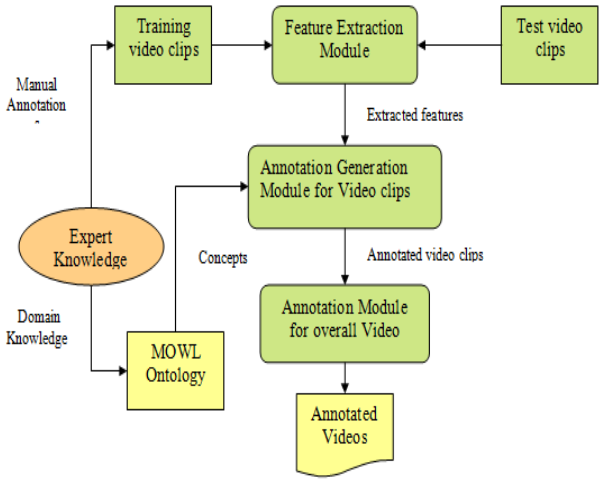

FIGURE 2.1: Framework for Conceptual annotation of videos and parts of videos.

framework for recommendation of videos. A **User Profile ontology** is created based on user attributes like age group, gender, etc. and using their preferences in movies. This ontology is then used for computing the recommendations detailed in Section 5. User profile ontology provides user preferences for the movies — in terms of concepts or classes in the movie ontology. A **MOWL parser** module parses the two ontologies and generates a Recommendation model (a Bayesian network) based on the user profile and associated movie concepts. This model is used to generate the initial movie recommendations for a user currently logged into the system. As the user selects movies to see, every click is recorded and fed back into the **Personalization module**. This module monitors the user preferences, and by using the most current movie preference of the user, tunes the initial recommendation model to generate personalized recommendation results. Over a period of time, recommendation improves as system dynamically tunes it according to the user preferences. The algorithm has been explained in section 5.3.

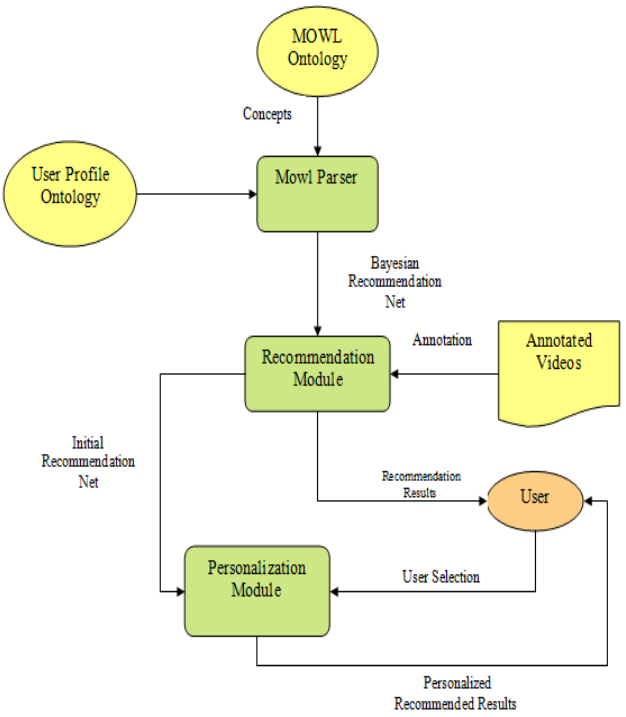

FIGURE 2.2: Framework for Personalized Recommendation of Videos

Chapter 3

Movie Domain

3.1 Movie Domain

We have tested our proposed framework for annotating and recommending the videos in the English movie domain. Different media patterns can be observed in different kinds of movies. These can help in classifying movies in many ways. For instance, a *romantic* scene has different sound (softer music) and visual (slower actor movements) features as compared to a fight scene in an *action* movie. So, we can say that specific media patterns which denote *action* genre can be observed more frequently in an *action movie* and same is true for other genres like *animation, romance* and *musical*.

We have used MOWL to encode the movie domain knowledge in a multimedia ontology. A snippet of the movie ontology created by domain experts is shown in figure 3.1. All important concepts related to a movie domain are taken into consideration. Concept `Movie` has related concepts like `Actor`, `Director`, `ReleaseDate`, `Genre`, and so on. The MOWL ontology also allows that specific media patterns which are related to a genre, can be attached as media properties with a genre node. An example is shown in figure 3.1 where Action properties (aural and visual properties) of media are associated as media features with Action concept in the ontology. These media patterns can be detected using media feature classifiers in order to help generate semantic annotation for the movie clips (refer section 4.3).

A part of information related to movies and other concepts in the ontology is collected from IMDB (Internet Movie Database)[1].Media feature based classifiers are used to perform movie genre classification on the basis of aural and visual

[1]http://www.imdb.com/title/tt1305591/.

features extracted from the media. Some examples of concepts and associated relations in movie ontology are given below:

- Movie hasGenre: Genre
- Movie hasActor: Actor
- Movie hasIMDBRating: IMDBRating
- Movie hasReleaseDate: Date

Movie is related to concepts like Actor, Director, Genre etc via properties. Media features associated with some genre related concepts are also shown in the ontology.

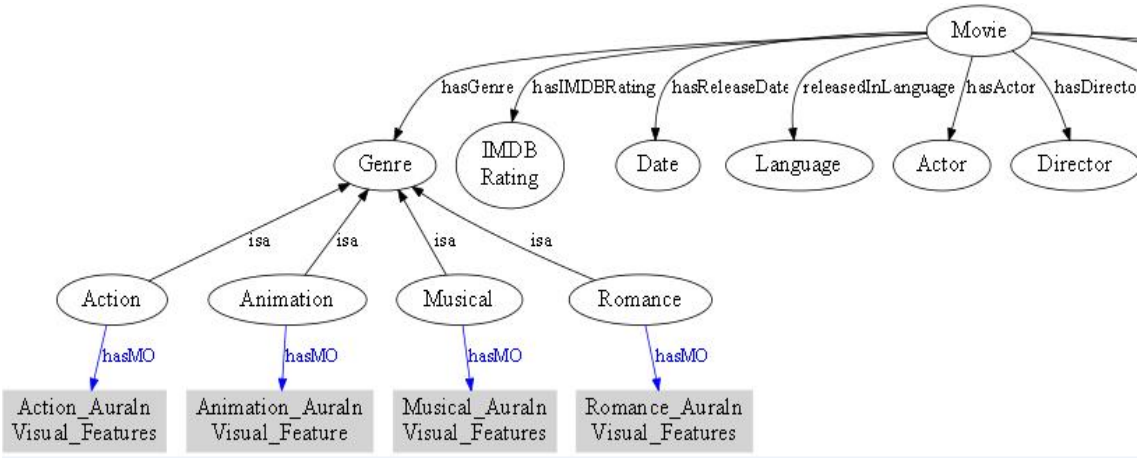

FIGURE 3.1: Snippet of Movie Ontology

Chapter 4

Conceptual Annotation

Conceptual annotations are done by extracting video scenes from the movie videos. These video clips are extracted manually from each movie and are further annotated by the system with a concept. Video clips are processed by system as groups of frames. Features from each group are extracted and then further used for video representation. Each video clip is represented in terms of an Indicator vector using the video description scheme proposed in [17]. These are then given as input to an SVM classifier for scene classification. The algorithms used for extracting video features are as follows:

4.1 Extraction of Visual Features

Visual features play a very important role in scene classification problem. Objects and scenes of an animation clip are quite different from an action scene. Also, most action scenes are quite fast in terms of shot changes as compared to romantic scenes. So, we have experimented with the following three visual features by processing 30% of each video.

4.1.1 Dominant Color Descriptor

To find out top-k dominant colors of a video, following steps are followed:

1. Divide video frames in S groups and compute number of frames to be processed from each group (30% of frames in a group).

2. For each group in a video:

   (a) For each frame in a group:

      i. Reduce the resolution of image frame (1/10 of original image). It has been verified that such operation is not affecting the accuracy of DCD computation.

      ii. Compute k-major dominant colors of that frame. To do this:

      iii. Change the color space of 256*256*256 colors into 25 major colors as given in [10]. This is done by computing the Euclidean distance between these 25 colors and each pixel color in an image. A pixel is now represented by a color out of these 25 colors, which is having minimum distance.

      iv. Now compute frequency of each of these 25 colors (count the number of pixels having a particular color)

      v. Assign the weights k, k-1, k-2, ...2,1 to the dominant colors. These weights represent the k-dominant colors of an image frame

   (b) After processing a group of frames, compute values for each of 25 colors by summing the weights corresponding to that color in the set of frames. Again top k colors form the Dominant Color Feature for that group.

3. After processing all groups, DCD features for each group are stored in the order: Most Dominant, second dominant and so on, and this dataset of size (S*k) will be used for clustering to compute Indicator Vector for a video.

4.1.2 Pace Characterization Features

The pace of a sequence is characterized by shot duration and number of shots. Earlier, we have experimented with Color Histogram difference for video segmentation [25]. But it was observed that edge detection techniques perform better for segmenting a video and determining shot change and length. So, Edge Detection technique is used to determine shot changes. We term a video as Fast paced video when we have a large number of shots of smaller duration, while in Slow paced videos shot lengths are pretty large. We classify shot duration into small, moderate and large using:

- small, if $shotlength < 60$
- moderate, if $60 \leq shotlength \leq 135$

- large, if $shotlength > 135$

Following Pace Characterization features of a video are computed for each group:

- cardinality of each type of shots: small, moderate and large
- normalized length component (nlc) for each type of shots using formula:
  nlc(small) = total small shot length/total no. of frames processed
  nlc(moderate) = total moderate shot length/total no. of frames processed
  nlc(large) = total large shot length/total no. of frames processed

The steps followed are as follows:

1. Divide video frames in S groups and compute number of frames to be processed from each group (30% of the frames in a group).
2. For each group in a video, segmentation is done as described in the following steps:
   (a) Find edges in two consecutive video frames using Canny Edge Detection method [9], as it makes the algorithm less sensitive to small changes.
   (b) Based on these edges, the sections (blocks) of video frames are compared to one another.
   (c) If number of different sections exceed a specified threshold, it is inferred that the scene/shot has been changed. (Threshold for experiments is taken as 80%)
   (d) Store the shot length and compute cardinality(small shots),cardinality(moderate shots) and cardinality(large shots)
   (e) When all frames of a group are processed, then compute nlc(small), nlc(moderate), nlc(large) using the above formula.
3. After processing all groups, Pace features for each group are stored in the order: cardinality(small shot), cardinality(moderate shot), cardinality(large shot), nlc(small), nlc(moderate), nlc(large) and this dataset of size (S*6) is used for clustering to compute Indicator Vector for a video.

4.1.3 SIFT Features

SIFT features of 30% consecutive frames from each group in a video are extracted. The steps are as follows:

1. Divide the video frames in S groups and compute the number of frames to be processed from each group.
2. For each group in a video:
   (a) For each frame in a group which are to be processed:
      i. Convert RGB to Gray Scale image for SIFT feature extraction.
      ii. Compute SIFT features from image using SIFT feature extraction algorithm [15]. It gives SIFT descriptors of length 128*N. This N can differ for different images. So we store each image descriptor as N*128 (descriptor') while preparing dataset for clustering.
   (b) Apply clustering and compute top-k representative SIFT features for a group.
   (c) Final Feature Vector for that group will be of size k*128 (k cluster centers) extracted in previous step.
3. After processing of all groups, SIFT features for each group are stored in the order: Most Dominant cluster center, second dominant cluster center and so on, and this dataset of size (S*k*128) will be used for clustering to compute Indicator Vector for a video.

4.2 Extraction of Aural Features

4.2.1 Pitch based Features

Pitch [1] is an important attribute in the analysis of speech signals.We have used RAPT algorithm [23] to determine pitch. Following Pitch based features [13] are extracted for each group in a video:

- Speech Ratio: ratio of length of speech frames to the entire audio clip.
- Pitch difference mean: mean of pitch differences between adjacent frames.

- Pitch difference standard deviation: standard deviation of pitch differences between adjacent frames.

The steps followed are as follows:

1. Divide the audio frames in S groups and compute number of frames to be processed from each group (30% of frames in a group/audio clip length).
2. For each group in audio following steps are performed:
   (a) Read audio frames from starting to last frame of a group
   (b) Calculate mean values for each frame(two channel values into single one)
   (c) Determine pitch using RAPT algorithm which gives Larynx frequency for each frame (or NaN for silent/unvoiced)
   (d) Normalize pitch vector using scaling formula so that range of all values of vector becomes 0 to 1.
   (e) Find number of elements in pitch vector which are above threshold. Threshold is taken as 0.3.
   (f) Compute Speech Ratio as ratio of number of elements greater than Threshold (Speech Frames) to total number of frames processed from a group.
   (g) Compute Pitch difference between adjacent rows and then Pitch difference mean and Pitch difference standard deviation.

   3. After processing all groups, Pitch features for each group are stored in the order: Speech Ratio, Pitch difference mean, Pitch difference standard deviation and this dataset of size (S*3) will be used for clustering to compute Indicator Vector for a video.

4.2.2 Volume Distribution Features

The volume distribution [13] of an audio clip reveals the temporal variation of signals magnitude, which is important for scene classification. Following Volume based features are extracted for each group in a video:

- rms value
- standard deviation

- mean
- volume dynamic range
- silence ratio
- zero crossing rate

The steps followed are as follows:

1. Divide audio frames in S groups and compute number of frames to be processed from each group (30% of the frames in a group/audio clip length).
   2. For each group, following steps are performed:
      (a) Read audio frames from starting to last frame of a group
      (b) Compute the magnitude of audio signal
      (c) Calculate y=mean values of each frame (two channel values into single one)
      (d) Calculate rms value(y), mean(y) , standard deviation(y) and $VolumeDynamicRange = (y_{max} - y_{min})/y_{max}$
      (e) Identify whether frames are voiced or not by checking condition: if $(framesignal_{value} - signal_{mean})/signal_{standardDeviation}$ is greater than threshold, then a frame is said to be voiced (1) otherwise it is unvoiced(0). After that count all the frames where voice=0 (silence).
2. Compute Silence Ratio = no. of Silence Frames/ total no.of processed frames Find zero crossing Rate = no. of zero crossings in a signal / total no. of frames in a signal.
3. After processing all the groups, Volume features for each group are stored in the order: rms value, standard deviation, mean, volume dynamic range, silence ratio and zero crossing rate and this dataset of size (S*6) will be used for clustering to compute Indicator Vector for a video.

4.2.3 MFCC based Features

10 MFCC features from each group in video are extracted. The steps followed are as follows:

1. Divide the audio frames in S groups and compute the number of frames to be processed from each group (30% of frames in a group/audio clip length).
2. For each group in audio following steps are performed:
   (a) Read the audio frames from starting to last frame of a group.
   (b) Calculate y=mean values of each frame (two channel values into single one).
   (c) Compute the 12 MFCC coefficients using algorithm [14] for each sample frame in audio clip. Parameters for algorithm are Hamming window in time domain, triangular shaped filters in mel domain, filters act in the absolute magnitude domain.
   (d) Reduce to 10 MFCC coefficients by applying Dimensional reduction using Principal Component Analysis [**?**].
   (e) Variance of these projected vectors forms the Feature Vector for that group.
3. After processing all groups, MFCC features for each group are stored and this dataset of size (S*10) will be used for clustering to compute Indicator Vector for a video.

4.3 Indicator Vector Generation

Here the proposed scheme uses a bag-of-words model where the words are visual words. Once we have K clusters in each feature space, for a video whose indicator vector is to be computed, S feature vectors in each of six feature spaces are computed as described in section 4.1. Each feature vector is then assigned to a particular cluster of its feature space by computing its distance from the cluster centers. The frequency of occurrence of a particular cluster is calculated for the given video. These frequencies form the indicator vector. So for six feature spaces and K clusters, an indicator vector of length $K \times 6$ is generated.

4.4 Annotation

4.4.1 Annotation of Movie Clips

Dataset consisting of different movie scenes, extracted from videos and represented as Indicator vectors, is prepared. These video clips are manually annotated by a domain expert. It was verified that no two video clips are same in the dataset. Since it is a classification problem, LIBSVM (Library for Support Vector Machines) implementation [4] of SVM Classification algorithm is used for classifying movie clips to different genre categories. Our problem is of multi-class classification (as there are more than 2 classes in our dataset). So we have used One-Versus-Rest implementation of LIBSVM for classifying video clips into their associated classes.

4.4.2 Annotation of Movies

After annotation of movie clips, genres for each movie video are predicted based on the following algorithm 1:

After annotation of movie clips, genres for each movie video are predicted based on algorithm 1. System determines count of each genre and total number of video clips annotated for a movie. It then computes percentage of each genre video with respect to total number of video clips.Then it checks the difference between percentage of occurrence of other genres and the most frequently occurring one, and if this difference is less than a threshold, it adds corresponding genres to final result set consisting of annotations for a movie.

Algorithm 1 Conceptual Annotation of Videos.

Input: *Annotated video clips of k movies*
Let $count_i$ = count of videos annotated with class(genre) i belonging to a movie k.
Let τ = count of classes(genres) into which video scenes can be classified.
Let P = percentage of class.
Let S = set of percentage of each class.
Output: $assign_k$ (set of genres assigned to a particular movie k) for each movie k.

for each input k **do**

  for each class i **do**
    Compute $count_i$
  end for
  $cg_k \leftarrow \sum_{i=1}^{\tau} count_i$

  for each class i **do**
    $P_i \leftarrow (counti/cg_k) * 100$
    add $S \leftarrow P_i$
  end for
  compute$\{max_i, P_{max_i}\} \leftarrow max\{S\}$
  $assign_k \leftarrow max_i$

  for each element e in $\{S\} - P_{max_i}$ **do**

    if $P_{max_i} - e < threshold$ **then**
      $assign_k \leftarrow e$
    end if
  end for
  Output $assign_k$
end for

Chapter 5

Movie Recommendation

5.1 Creation of user profile Ontology

Based on a survey conducted for identifying the movie preferences of people belonging to different age groups and gender, we created a User Profile ontology. We provided a list of some popular movies with their description to 100 users and asked them to rate it over a scale of 3, based on how interested they are in watching these movies. We also asked some other questions like their age, gender, profession and favorite Eglish movie actors. Based on the data collected from survey, we created a User Profile Ontology as shown in figure 5.1. Some examples of relations and associated concepts in this ontology are:

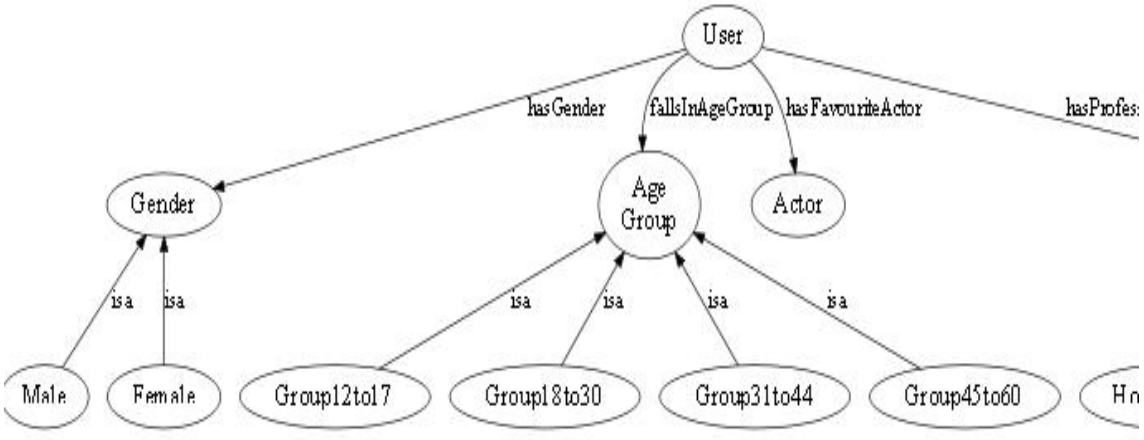

FIGURE 5.1: Section of User Profile Ontology

- User hasGender: Gender

- User hasAgeGroup: AgeGroup
- User hasFavoriteActor: Actor

Figure 5.2 shows a code snippet of User Profile ontology:

```
<owl:ObjectProperty rdf:ID="hasGender">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="User"/>
    <rdfs:range rdf:resource="Gender"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="fallsInAgeGroup">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="User"/>
    <rdfs:range rdf:resource="AgeGroup"/>
</owl:ObjectProperty>

<owl:ObjectProperty rdf:ID="hasFavouriteActor">
    <rdf:type rdf:resource="&owl;FunctionalProperty"/>
    <rdfs:domain rdf:resource="User"/>
    <rdfs:range rdf:resource="Actor"/>
</owl:ObjectProperty>

<owl:Class rdf:ID="Actor" />
<owl:Class rdf:ID="Gender" />
<owl:Class rdf:ID="Male">
    <rdfs:subClassOf rdf:resource="Gender"/>
</owl:Class>
<owl:Class rdf:ID="Female">
    <rdfs:subClassOf rdf:resource="Gender"/>
</owl:Class>

<owl:Class rdf:ID="AgeGroup" />
<owl:Class rdf:ID="Group12to17" >
    <rdfs:subClassOf rdf:resource="AgeGroup"/>
```

FIGURE 5.2: code snippet of User Profile ontology

5.2 Linking User Profile to Movie Ontology

We found by analysing the survey that users belonging to a particular profile have similar interests in movies. Thus attributes like gender, age-group, favorite actor, etc. can help decide initial recommendation to a user according to his or her profile. In this research work, we have focused mainly on recommendation of movie videos based on age and gender information of a user. The user nodes in the User Profile ontology are linked to movie genres in the Movie ontology using a `hasRec` relation.

With each of these nodes, a conditional probability table (CPT) is associated that stores the prior probabilities which help in initial recommendation. These are as shown in figure 5.3. We computed these prior probabilities from the user survey conducted.

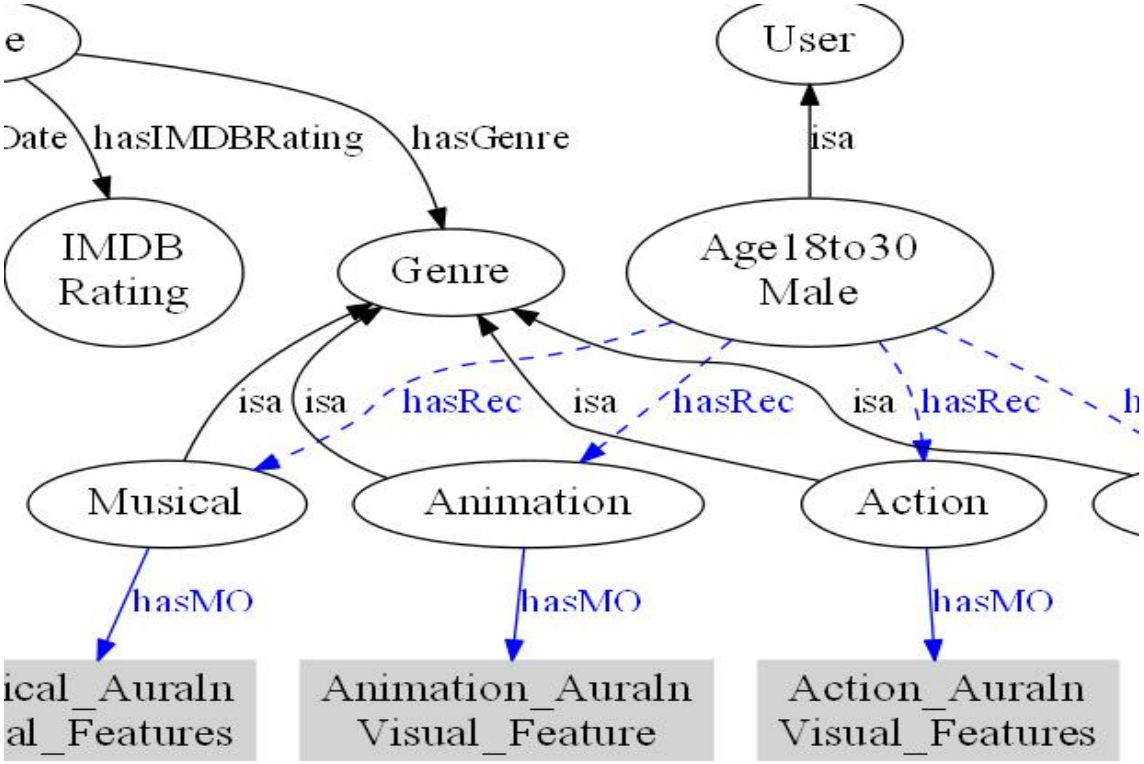

FIGURE 5.3: Linking User Nodes to Movie Ontology

MOWL snippet depicting the linking of ontology nodes for recommendation process is given in figure 5.4.

5.3 Movie Recommendation System

Our recommendation system works by finding the likelihood of movie genres a user prefers to see. It uses a probabilistic reasoning scheme to determine the ranking of a movie which a user may prefer. User profile and Movie ontology provides other related concepts like movie ratings,actors etc. Our system utilizes this knowledge to compute final ranking of movies and then shows recommendations to a user.

```
<mowl:propagateMedia rdf:ID="hasRec">
    <rdfs:label></rdfs:label>
    <rdfs:domain rdf:resource="Age31to44Female"/>
    <rdfs:range  rdf:resource="Action"/>
    <rdfs:comment> Genre determines the User behavior</rdfs:comment>
</mowl:propagateMedia>

<mowl:propagateMedia rdf:ID="hasRec">
    <rdfs:label></rdfs:label>
    <rdfs:domain rdf:resource="Age45to60Male"/>
    <rdfs:range  rdf:resource="Action"/>
    <rdfs:comment> Genre determines the User behavior</rdfs:comment>
</mowl:propagateMedia>

<mowl:propagateMedia rdf:ID="hasRec">
    <rdfs:label></rdfs:label>
    <rdfs:domain rdf:resource="Age45to60Female"/>
    <rdfs:range  rdf:resource="Action"/>
    <rdfs:comment> Genre determines the User behavior</rdfs:comment>
</mowl:propagateMedia>


    <mowl:propagateMedia rdf:ID="hasRec">
    <rdfs:label></rdfs:label>
    <rdfs:domain rdf:resource="Age12to17Male"/>
    <rdfs:range  rdf:resource="Animation"/>
    <rdfs:comment> Genre determines the User behavior</rdfs:comment>
</mowl:propagateMedia>


    <mowl:propagateMedia rdf:ID="hasRec">
```

FIGURE 5.4: MOWL Snippet linking User Profile to Movie Ontology

5.3.1 Recommendation Model Generation

Based on the media patterns detected in a movie, the media nodes in the Movie ontology are instantiated and are further used for predicting movie genre preferences. For the user currently logged in, system has the following information: age, gender and favorite actors (optional). Based on this user profile, system generates a Recommendation model from the two ontologies. This is organized as a Bayesian tree. A Bayesian net [7] represents conditional dependencies between concepts or nodes. An example of a Recommendation net generated for a user with profile — Male of age group 12 to 17 years — is shown in figure 5.5. Here the user profile is shown as an ellipse, and the movie genres as gray boxes. The conditional dependencies between the nodes are are shown as a pair $P(G \mid U)$ $P(G \mid \not{U})$, where G denotes movie genre and U denotes user profile. The probabilties shown along

with a concept in the Bayesian tree, denote the initial probabilities set with each node, as extracted from the MOWL ontology, which allows encoding of CPTs.

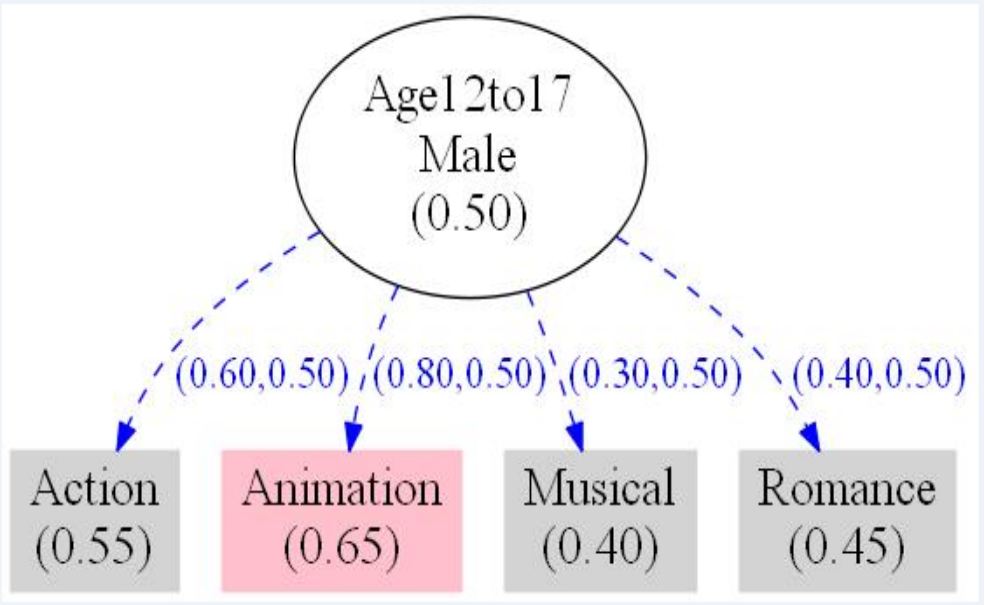

FIGURE 5.5: Recommendation Model extracted from Ontology as Bayes Tree.

5.3.2 Initial Recommendation

Our recommendation module uses initial CPTs, encoded in the ontology, for computation of posterior probabilities which determine whether a new user will prefer to see a movie or not. As we can see from the figure, the ontology has the knowledge that a typical 12 to 17 year old male prefers animation movies. This is reflected in the 0.80 conditional probability for $P(G = Animation \mid U = (12to17yearoldMale))$ in the figure. Thus the `Animation` genre node has a prior probability of 0.65 which is higher than a threshold and thus is shown in pink. To determine the recommendation, when a movie or a movie scene, which is annotated with a particular *genre* is to be ranked, the corresponding *genre* node in the Recommendation model is instantiated, and belief propagation is done. The posterior probability of the root node which is the *User profile* node, gives the recommendation measure of the movie. Thus the CPTs associated with the nodes determine the initial recommendation for the user when he first logs into his account. This is how our recommendation system handles the initial cold-start problem.

First n best-ranked movies, as indicated by posterior probability of the user node, are then placed in a ranked list for recommendation. An example is shown in

figure **??**. When a movie or clip which is annotated to be of the *Animation* genre, is given, the posterior probability which is computed after taking the relevant evidence into account is 0.72. This provides the ranking for the movie, and as it is quite high, the Animation movie is recommended to the user — which is a good recommendation considering his user profile interest, as encoded in the ontology.

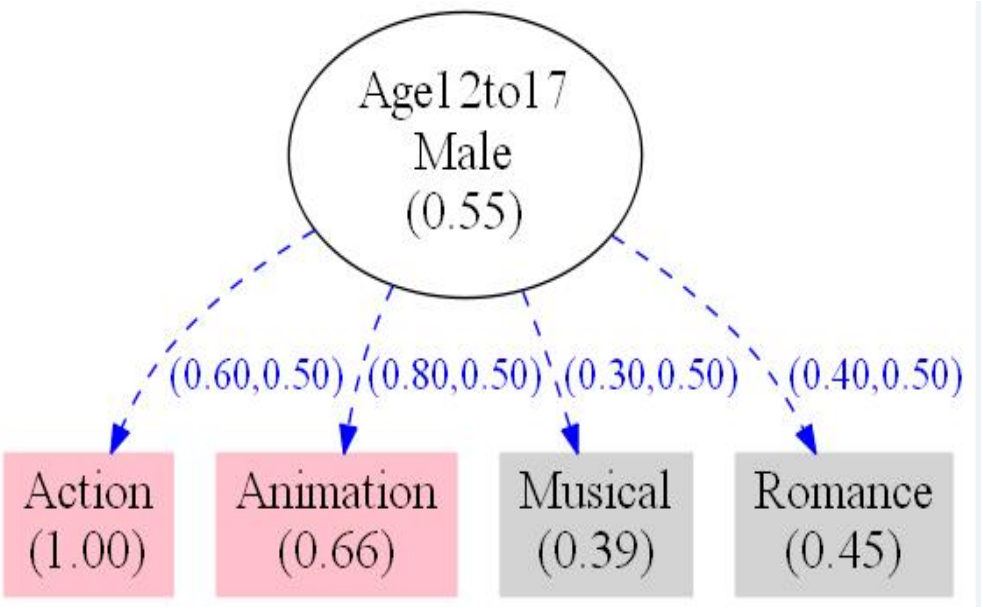

FIGURE 5.6: Posterior Probability computation for Action Movie

5.3.3 Personalization

The personalization module stores a copy of initial Bayesian network for each user. It monitors the user clicks and by using current preference of a user, modifies the CPTs of initial Bayesian network and then uses this modified net for further recommendation process. The novelty in our approach is that we have tried to adjust the CPTs of Bayesian network so that system can accordingly adapt to user preferences.

We have experimented with Counting Learning methodology using NeticaJ [1] for changing the CPTs to account for current preference of a user. The methodology works as follows:

At each node, we set one experience number for each possible configuration of states of parent nodes and its associated probabilities. The experience value denotes the number of cases that have been seen. By setting experience value as k, it means that the system has learnt the given CPTs from k cases. When a user

[1] http://www.norsys.com/neticaj/docs/NeticaJ_Man.pdf.

clicks on a new item, system stores it as a case file, by setting nodes corresponding to the selected item in initial Bayes Net as true and other nodes as false. This case data represents current preference of a user. An example is shown in figure 5.7 representing a case in which user belongs to 12 to 17 age group, gender male and has clicked on a Romantic movie to watch. User node (Age12to17Male) and Romance genre node are set to true and other nodes in Recommendation net are set to false.

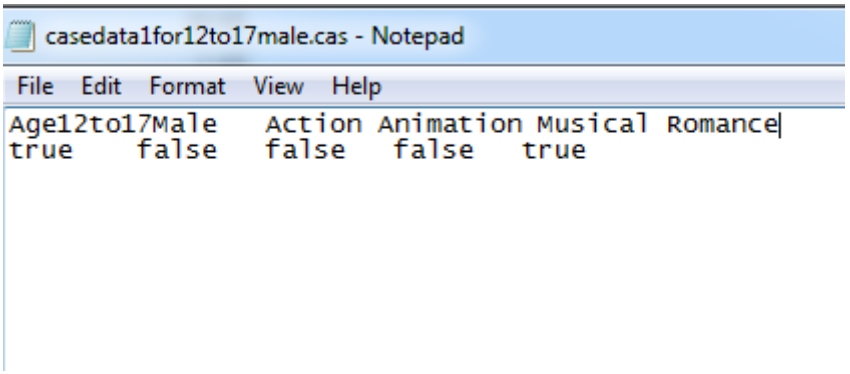

FIGURE 5.7: Example of a Case File

A degree is also associated with each case file, which we have chosen as 1 as system is learning one user case at a time.System adjusts CPT of only those nodes for which the case supplies a value (finding).The new experience number is found from the old one by using following formula:

$$Expr_{new} = Expr_{old} + degree$$

For computation of probability for node state that is consistent with the case, following formula is used :

$$Probability_{new} = (Probability_{old} * Expr_{old} + degree)/Expr_{new}$$

The probabilities of other nodes are changed by:

$$Probability_{new} = (Probability_{old} * Expr_{old})/Expr_{new}$$

After changing Conditional Probability Tables, Bayesian net is then used by recommendation module for computation of posterior probabilities. Movies that are best ranked, as indicated by posterior probability of user node and according to other related concepts in the ontology, are then selected for recommendation.An

example illustrating how a Personalized Net changes on the basis of User selection has been shown in figure 5.8. Initially Recommendation net has Conditional Probability Tables associated with each node. In figure, CPT associated with Animation genre node has 0.80 conditional probability for $P(G = Animation \mid U = (12to17yearoldMale))$ in the figure. Thus the Animation genre node has prior probability of 0.65 which is highest among all the genres. So, when a user logs in for the first time, movies having Animation genre will be ranked higher than all other genre movies. Then system tunes the CPTs by monitoring the user clicks. When user selects an Action movie, system increases the probability for $P(G = Action \mid U = (12to17yearoldMale))$ to 0.68 and decreases the probability of other nodes as shown in the figure. Now the prior probability for Action genre increases and other genre decreses, so movies having Action genre are ranked more higher than other genre movies. This is how ranking of movies is changed as user selection changes. System also takes care of mixed genres, for instance, if a user likes movies having Music and Romance genre, then it ranks movies which are associated with both the genres higher than other movies.

5.3.4 Personalized Recommendation Algorithm

Algorithm 2 explains how recommendation system works for a user currently logged in. System generates initial Algorithm 2 explains how recommendation system works for a user currently logged in. System generates initial recommendations when a user logins for the first time, then personalizes the Recommendation net for a user for generating recommendations. System instantiates user node in User profile ontology that corresponds to current user profile, then recursively extracts all the concepts of Movie ontology that are linked to user profile and creates Bayes Net with these extracted concepts. Based on the concepts (genre) identified in a movie, system computes the posterior probability of a movie for a User and recommends the best ranked movies to a user. System personalizes the recommendations by writing a case file for current selection of user. It then modifies existing CPTs in Bayesian Network as explained in section 5.3.3. System then again computes the posterior probability of each movie for current user, ranks the results and generates recommendations.

Algorithm 2 Personalized Recommendation of Videos.

Input: *Movie Ontology encoded in MOWL, Set of Annotated Movies M, User Profile U.*
Let G = set of class(genre) g belonging to a particular movie m.
Let *node* = concept in ontology corresponding to the user profile U.
Let *isLink(x,y)* returns true if x is related to y by a recommendation link.
Let S = set of *nodes*.
Let B_{net} be the Bayesian Net consisting of S and U.
Let *Posterior(x)* gives the posterior probability at node x.
Let A = set of posterior probabilities initially empty.
Let C = set of movies selected by user in an order of his selection.
Let *writeCase(x,y)* stores the case for a node as true if its finding is true otherwise false.
Let d = number of cases learned at a time.
Let $Expr_{old}$ = experience value denoting the number of cases that have been seen
Let $prob_{n_{old}}$ = probability in CPT for the node state
Output: Recommendations according to posterior probabilities in Set A.

for each input m ∈ M **do**

  for each genre g ∈ m **do**
    $G \leftarrow g$
  end for
  Instantiate $node \widehat{=} U$
  Extract from Ontology recursively
  $S \leftarrow c, x \mid isLink(node, c) \cup isLink(c, x)$
  Construct $B_{net} \leftarrow \{S\} \cup \{U\}$

  for each genre g ∈ G in B_{net} **do**
    $belief_g \leftarrow true$
  end for
  Compute $p_m \leftarrow Posterior(U)$
  $A \leftarrow A \cup p_m$
end for
Show Recommendations according to posterior probabilities ∈ A

for each input J ∈ C **do**

  for each genre g ∈ G in B_{net} **do**
    if g ∈ J **then**
      writeCase(g,true)
    else
      writeCase(g,false)
    end if
  end for
  $Expr_{new} \leftarrow Expr_{old} + d$

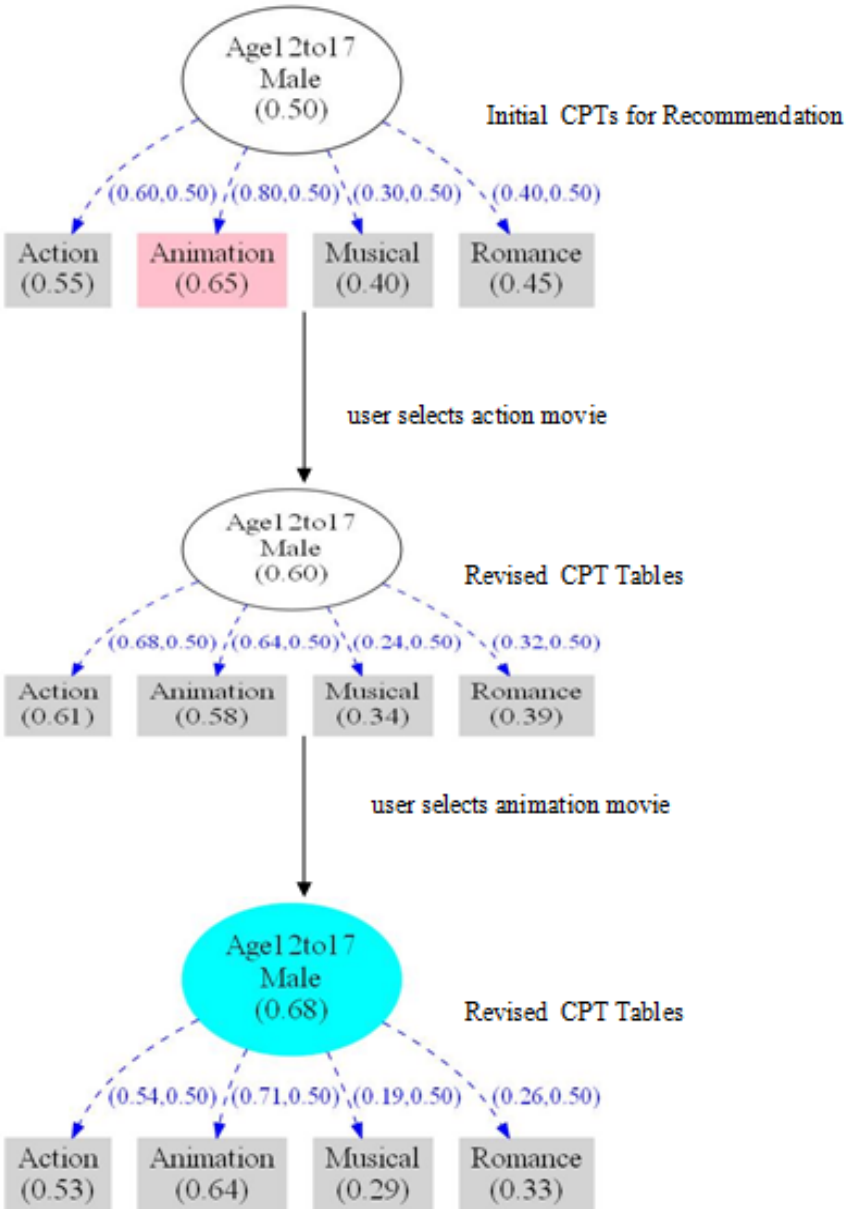

FIGURE 5.8: Example illustrating how Personalized Net changes upon User Selection.

44:
for each node n $\in B_{net}$ **do**
46: **if** $case_n$ == true **then**
$prob_{n_{new}} \leftarrow prob_{n_{old}} * Expr_{old} + d)/Expr_{new}$
48: **else**
$prob_{n_{new}} \leftarrow prob_{n_{old}} * Expr_{old})/Expr_{new}$
50: **end if**
end for
52:
for each input m $\in$ M **do**
54: Repeat Steps 21 to 25
end for
56: Show Recommendations according to posterior probabilities
end for

5.3.5 Revision of Initial CPTs over a period of time

Probabilities in initial CPTs can be revised over a period of time using information stored in case files. Such CPTs then represent general changes in user model for a set of similar users that used the system over a period of time. Thus personalized recommendation nets are used to change the CPTs stored in the User profile ontology to reflect changing preferences of a User profile.

Chapter 6

Experimental Results

6.1 Conceptual Annotation

6.1.1 Annotation of Movie Video Clips

The dataset contains 317 movie clips with their Indicator Vector representation. Movie clips consist of Action, Animation, Musical and Romance scene clips from different movies.It has been verified that no two video clips are same in our dataset.The statistics of video clips has been given in Table 6.1.

Movie Genre	No. of video clips
Action	113
Animation	47
Musical	76
Romance	81

TABLE 6.1: Video Clips Statistics

We have used One-Versus-Rest implementation of LIBSVM. 70% of videos in our dataset are taken for training the system and rest of them have been considered for test set. These training and test sets are converted into LIBSVM format and then provided as input to Classifier. Training of classifier is done using 6-fold Cross Validation.Results are shown in table 6.2 and table 6.3.

Number of videos in training set: 224
Number of videos in test set: 93

Results
Classification Accuracy = 76.3441%

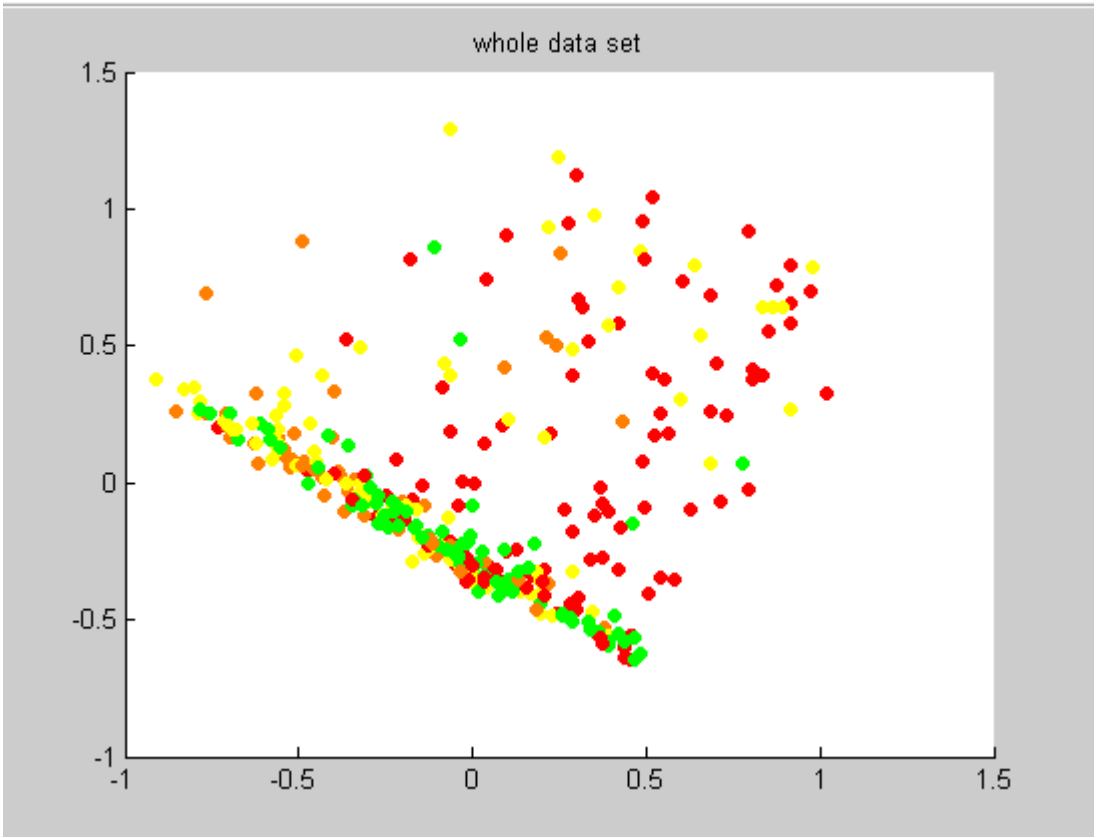

FIGURE 6.1: Plotting of all data points

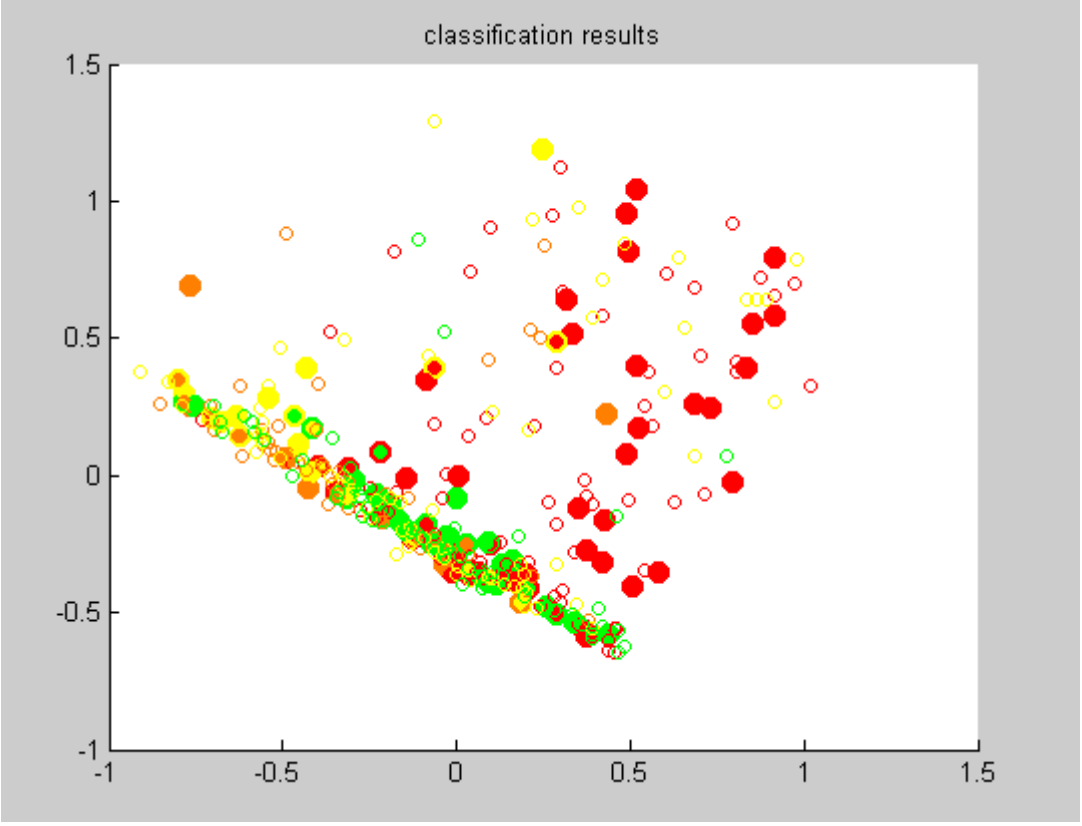

FIGURE 6.2: Classification Results.

Movie Genre	Action	Animation	Musical	Romance
Action	34	1	3	3
Animation	1	8	1	0
Musical	2	2	9	1
Romance	2	4	2	20

TABLE 6.2: Confusion Matrix

Movie Genre	Precision	Recall	F-Score
Action	0.8718	0.8293	0.8500
Animation	0.5333	0.8000	0.6400
Musical	0.6000	0.6429	0.6207
Romance	0.8333	0.7143	0.7692

TABLE 6.3: Precision, Recall and F-Score

6.1.2 Annotation of Movies

We have chosen movie videos as dataset for experiments. Four movie genres are taken into consideration, namely Action, Animation, Musical and Romance. Our system is designed such that it does automatic scene classification of movie clips into above genres, and based on that it predicts genres associated with a movie by assigning the genres if they occur above a threshold count.

In all experiments, it is assumed that movie videos consist of only above genres, so the dataset taken into consideration only consists of video clips belonging to only one of these genres.

Our database of movies comprises a total of 72 movies, belonging to either of the above, or a mix of one or more genres. The statistics has been given in table 6.4

Movie Genre	No. of Movie videos
Action	29
Action, Romance	02
Animation	07
Animation, Musical	04
Animation, Musical, Romance	01
Musical	08
Musical, Romance	03
Romance	18

TABLE 6.4: Movie Videos Statistics

After annotation of movie clips, we predicted the genres associated with a movie based on algorithm 1 on page 17.For this algorithm, we set the threshold as 15%.

After this, we curated movies in our database by verifying genres assigned to them by above algorithm with that given on IMDB.

6.2 Recommendation

We have experimented our recommendation approach on Movie dataset prepared as described in Section 5.Dataset comprises 72 movie videos as well as 317 movie scenes extracted from them classified into different categories.

We have created a prototype recommendation system and implemented a demo website designed using J2EE. We have used MowlParser in the implementation of our algorithm, which uses nanoxml for parsing the ontology and NeticaJ for Bayesian reasoning.

The UI Screen is shown in the figure 6.3. When a user logs in to the website, he provides initial information (age, gender, favorite actor) to the system. Initial recommendations are then shown to user on the main screen. These recommendations are explained by ranking computed using posterior probabilities and other related information in the ontology. Another option which is given along with the recommendations is to see another list of movies from which a user can select if he doesn't like top-ranked videos recommended by the system.

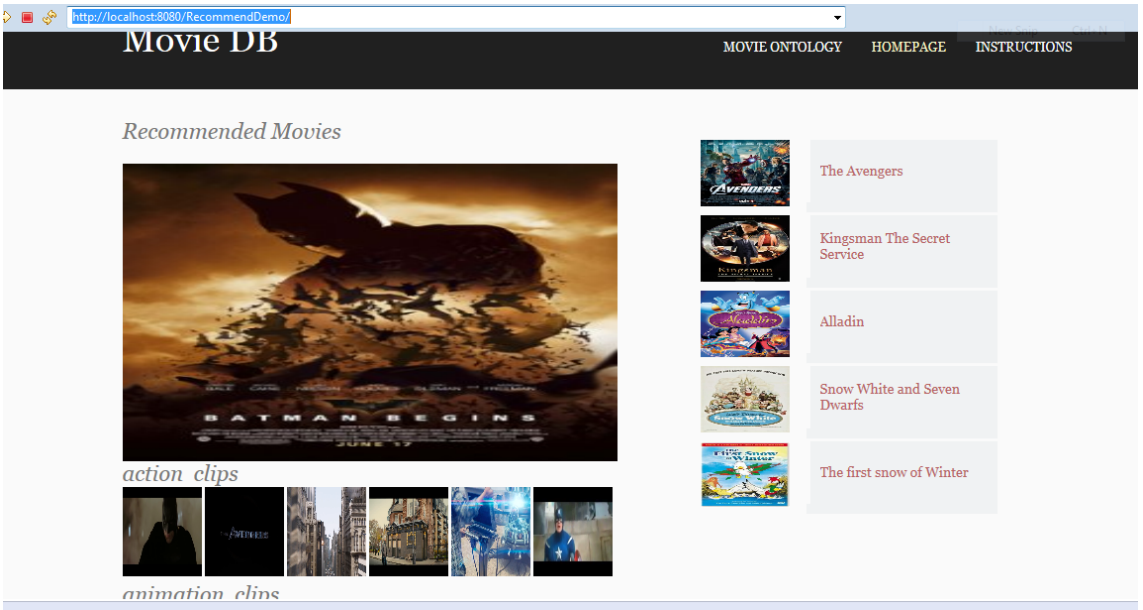

FIGURE 6.3: UI Screen for Movie Recommendation

The web application was given to 7 users (having different profiles) along with a list of movies which we have taken in dataset in order to make him more informed about movie options from which he can choose.

Table 6.5 presents for a single user, how the posterior probabilities computed by the algorithm and ranking changes according to user selection.

		Initial		Selection 1 Action		Selection 2 Animation		Selection 3 Animation	
S No	Genre	Ranking	Probabilities	Ranking	Probabilities	Ranking	Probabilities	Ranking	Probabilities
1	Action	2	0.54	1	0.67	2	0.69	2	0.71
2	Animation	1	0.61	2	0.65	1	0.75	1	0.81
3	Musical	6	0.37	6	0.41	6	0.44	6	0.47
4	Romance	5	0.44	4	0.48	5	0.52	4	0.54
5	Action, Romance	3	0.48	3	0.56	3	0.54	5	0.50
6	Animation, Musical	4	0.48	5	0.47	4	0.53	3	0.57
7	Animation, Musical, Romance	7	0.43	7	0.37	7	0.37	7	0.36
8	Musical, Romance	8	0.32	8	0.31	8	0.29	8	0.26

TABLE 6.5: Posterior Probabilities and Ranking for a User based on his selections

We have evaluated our recommender system's performance based on following measures:

1. Accuracy: User selections has been recorded and checked if he is selecting a movie from main recommendation list or from a list of other movies. Then, accuracy is calculated using:

$$Accuracy = k/(k+m)$$

where k=number of items selected from recommended list, and m=number of items selected from a list of other movies.

Recommendation accuracy for 7 subjects has been computed and given in table 6.6. It is evident from the results that on an average, 69% of the recommended items are correct and are in accordance with the user interests.

We have compared recommendation accuracy of our system against K-Nearest Neighbour (K-NN) approach proposed in [24]. As shown in figure 6.4, accuracy for MOWL-based framework is higher as such system is also considering the similarities among items along with ratings and other factors while generating recommendations. K-NN approach doesn't work well when the number of users are

Users	Accuracy
User 1	0.83
User 2	0.50
User 3	0.83
User 4	0.67
User 5	0.50
User 6	0.83
User 7	0.67
Average	0.69

TABLE 6.6: Recommendation Accuracy

very small. Afterwards, as the number of users increases, accuracy for both the techniques also increases.

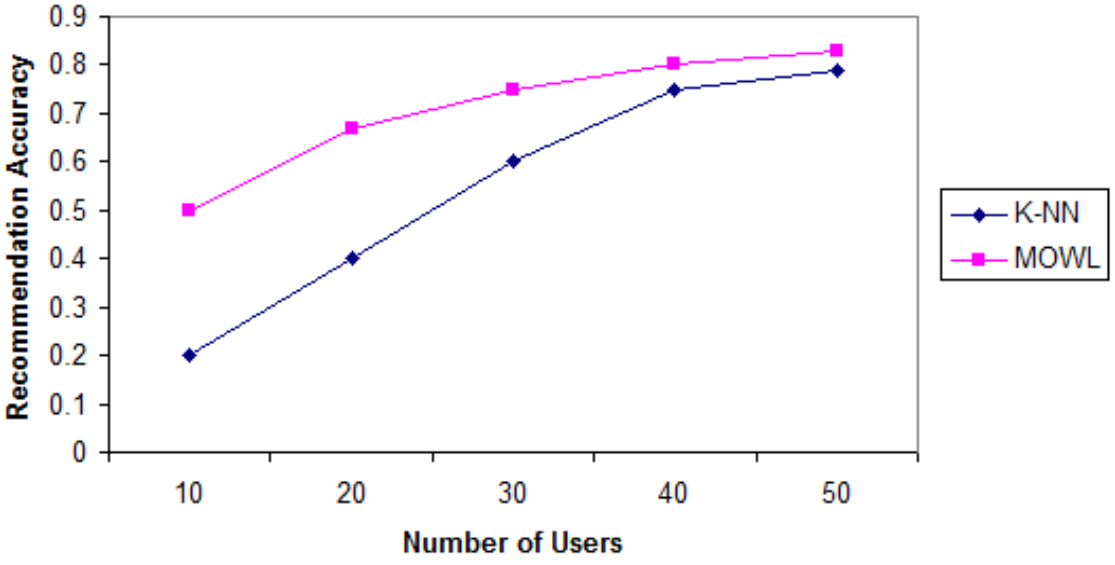

FIGURE 6.4: Accuracy (MOWL vs K-NN approach)

We have also evaluated our algorithm in terms of precision and recall for varying lengths of recommendation list. A good recommendation system has high precision and recall values.

$$Precision = t_p/(t_p + f_p)$$

$$Recall = t_p/(t_p + f_n)$$

where t_p denotes number of relevant items that are recommended, f_p denotes irrelevant items that are recommended and f_n denotes relevant items that are not recommended. Items rated as 4 or 5 are considered as relevant to the user and items which are rated below 3 are considered as irrelevant. Figures 6.5 and 6.6 show the precision and recall as a function of number of recommendations. We have compared precision and recall results for our recommendation system

against K-NN approach. It is quite clear from the graphs that precision and recall improves with increase in the number of recommendations. Also, these values are quite high when generating recommendations using MOWL based framework, as compared to the K-NN approach. Figures also show that initially MOWL based recommendation results are better as recommendations are genreated based on similarity among movies. It doesnot depend on the factors like number of users, number of ratings given by a user etc, as in K-NN approach.

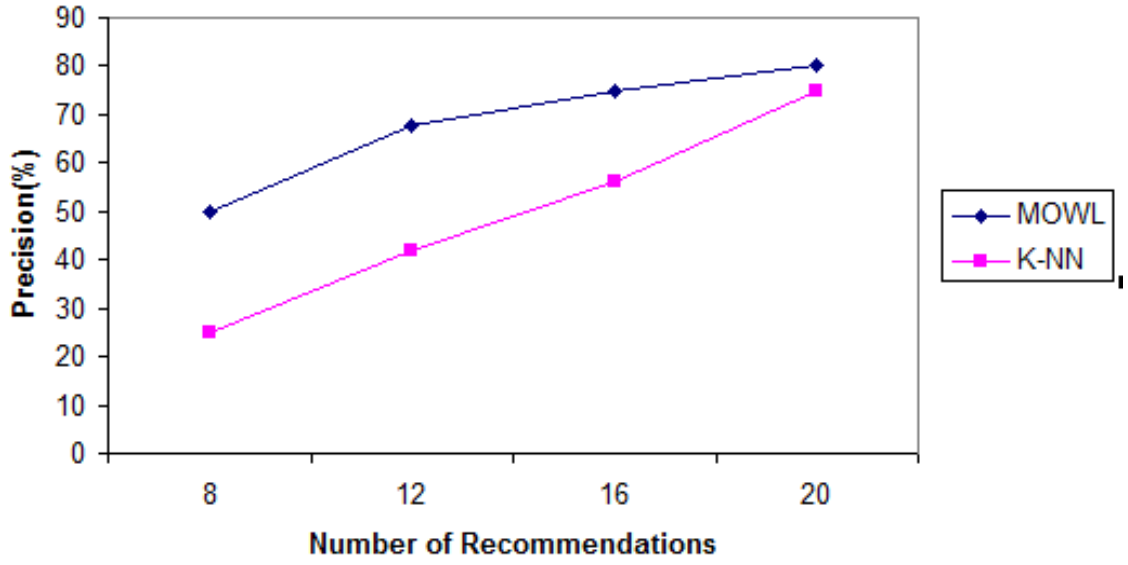

FIGURE 6.5: Precision vs No. of Recommendations

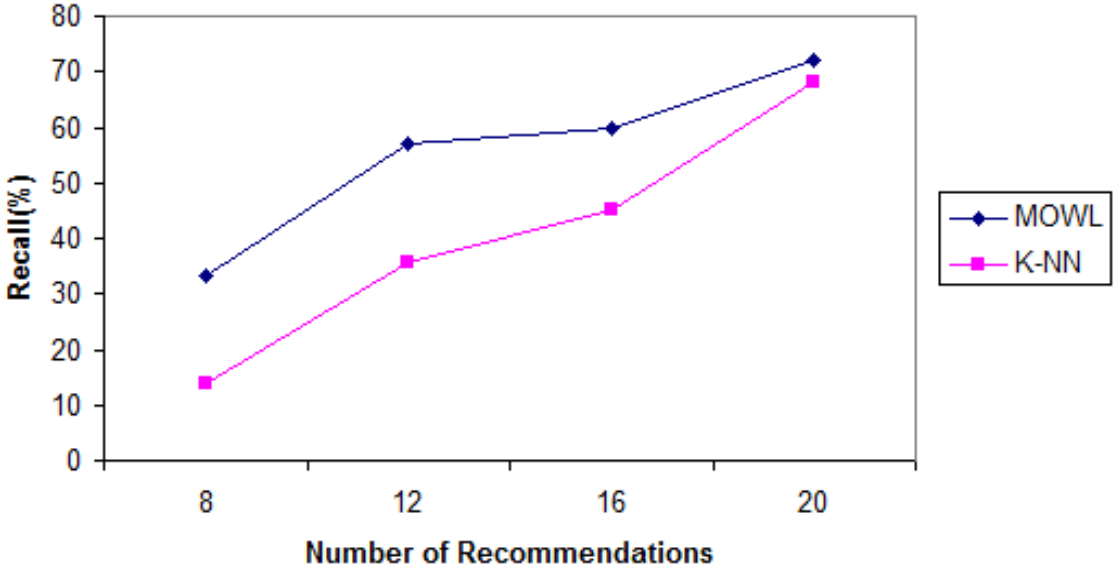

FIGURE 6.6: Recall vs No. of Recommendations

Kendall Rank Correlation Coefficient: To evaluate the ranking of items in recommendation list, we have asked users to rank items according to them and then computed association between the two rankings using Kendall rank correlation coefficient. It is a measure of rank correlation, i.e., the similarity of orderings of the data when ranked by each of the quantities. Kendall Rank Coefficient computed for 7 subjects are given in table 6.7. Coefficient values are computed for initial recommendations and for the list of movies recommended after next two selections made by him. On an average, coefficient value for each user is more than 0.6, which indicates that the order of ranked videos is quite appropriate and in accordance with the user preferences. Average coefficient value is 0.69 which shows that users are quite satisfied by the order in which videos are ranked and shown to them by the system.

$$\tau = (pairs_{concordant} - pairs_{discordant})/(0.5 * n * (n-1))$$

where, n=total number of ranked items, $pairs_{concordant}$ = number of concordant pairs and $pairs_{discordant}$ = number of discordant pairs.

Users	Initial	After Click 1	After Click 2	Average (Tau)
User 1	0.60	0.87	0.73	0.73
User 2	0.73	0.33	0.73	0.59
User 3	0.47	0.87	0.60	0.65
User 4	0.73	0.60	0.73	0.69
User 5	0.87	0.73	0.73	0.78
User 6	0.33	0.47	0.73	0.51
User 7	0.73	0.87	0.73	0.78
Average				0.67

TABLE 6.7: Kendalls Tau Coefficient

Chapter 7

Conclusion

We have presented a probabilistic approach for personalized recommendation using multimedia ontology. We have proposed an approach for generating semi-automated, semantic annotation of multimedia segments by using a machine-learning technique based on media feature extraction. Using this framework, video segments as well as full videos can be recommended to a user based on his/her interests. We have performed experiments on an English movie dataset, tested the performance of our system, and validated the efficiency of our approach. We can improve the system performance by increasing the number of users in our experiments, employing more user-attributes in profiling, and validating the recommendations made to them. Adding more movie genres and a variety of media features will further enhance the efficiency of this recommendation system. Future work for this research also includes validating our approach by comparing our results with other movie recommendation systems.

Bibliography

[1] A.R. Abu-El-Quran and R.A Goubran. Pitch-based feature extraction for audio classification. *In Proc., International Workshop Haptic, Audio, Visual Environments and their Applications, Ottawa, ON, Canada*, pages 43–47, September 2003.

[2] Stuti Ajmani, Hiranmay Ghosh, Anupama Mallik, and Santanu Chaudhury. An ontology based personalized garment recommendation system. *IEEE/WIC/ACM International Conferences on Web Intelligence (WI) and Intelligent Agent Technology (IAT)*, 2013.

[3] J. (Ed.) Allan, J. Aslam, N. Belkin, C. Buckley, J. Callan, W. B. (Ed.) Croft, S. Dumais, N. Fuhr, D. Harman, D. Harper, D. Hiemstra, T. Hofmann, E. Hovy, W. Kraaij, J. Lafferty, V. Lavrenko, D. Lewis, L. Liddy, R. Manmatha, A. Mccallum, J. Ponte, J. Prager, D. Radev, P. Resnik, Rosenfeld R. Roukos S. Sanderson M. Schwartz R. Singhal A. Smeaton A. Turtle H. Robertson, S., E. Voorhees, R. Weischedel, J. Xu, and C. Zhai. Challenges in information retrieval and language modeling. *SIGIR Forum 37*, 37(1):31–47, 2003.

[4] C.-C. Chang and C.-J. Lin. Libsvm : a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2011.

[5] Y. Deng, Z. Wu, C. Tang, H. Si, H. Xiong, and Z. Chen. A hybrid movie recommender based on ontology and neural networks. *Proc. International Conference on Green Computing and Communications & International Conference on Cyber, Physical and Social Computing (GREENCOM / CPSCOM 10), Washington DC, USA, IEEE Computer Society*, pages 846–851, 2010.

[6] E. A. Eyjolfsdottir, G. Tilak, and N. Li. Moviegen: A movie rec system. *IEEE Trans on Mult*, 15(5), AUGUST 2010.

[7] David H. A tutorial on learning with bayesian networks. *Learning in Graphical, M.J. Models, ed, MIT Press, Cambridge,1999. Also appears as Technical Report MSR TR-95-06, Microsoft Research*, MARCH 1995.

[8] Kazuhiro Iwahama, Yoshinori Hijikata, and Shogo Nishida. Content-based filtering system for music data. *Symposium on Applications and the Internet-Workshops. IEEE Computer Society Press, Tokyo, Japan*, 2004.

[9] JohnCanny. A computational approach to edge detection. *In IEEE Transactions On Pattern Analysis And Machine Intelligence*, 8(6), 1986.

[10] K.C.Ravishankar, B.G.Prasad, S.K.Gupta, and K.K.Biswas. Dominant color region based indexing for cbir. *In International Conference on Image Analysis and Processing, Proceedings published by IEEE CS Press*, September 1999.

[11] Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. Applying collaborative filtering to usenet news. *Communications of the ACM*, 40(3):77–87, March 1997.

[12] Freddy Lecue. Combining collaborative filtering and semantic content-based approaches to recommend web services. *IEEE Fourth International Conference on Semantic Computing*, 2010.

[13] Zhu Liu, Jincheng Huang, and Yao Wang. Audio feature extraction and analysis for scene classification. *in IEEE Signal Processing Soc. 1997 Workshop Multimedia Signal Processing*, 1997.

[14] Beth Logan. Mel frequency cepstral coefficients for music modelling. *In ISMIR*, 2000.

[15] D. Lowe. Distinctive image features from scale-invariant keypoints. *In International Journal of Computer Vision*, 20:91–110, 2003.

[16] Anupama Mallik, Hiranmay Ghosh, Gaurav Harit, and Santanu Chaudhury. Mowl: An ontology representation language for web based multimedia applications. *ACM Trans Multimedia Computing, Communications and Applications, (in press).*

[17] Anupama Mallik, Ankur Jain, Mansi Matela, Santanu Chaudhury, and Sukumar Bhattacharya. Multimedia ontology learning for automatic annotation and video browsing. *Proceedings of the 1st ACM international conference on Multimedia information retrieval*, 2008.

[18] N. G. Meshram and A. P. Bhagat. Connotative feature extraction for movie recommendation. *International Journal of Image Processing (IJIP)*, 8(5), 2014.

[19] Jochen Nessel and Barbara Cimpa. The movieoracle - content based movie recommendations. *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, 2011.

[20] D. H. Park, H. K. Kim, I. Y. Choi, and J. K. Kim. A literature review and classification of recommender systems research. *Expert Systems with Applications*, 39:10059–10072, 2012.

[21] MICHAEL J. PAZZANI. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*, 13:393–408, 1999.

[22] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th international conference on World Wide Web*, 2011.

[23] D. Talkin. A robust algorithm for pitch tracking (rapt). *In Speech Coding and Synthesis*, 1995.

[24] M. Vozalis and K. G. Margaritis. Collaborative filtering enhanced by demographic correlation. *In AIAI symposium on professional practice in AI, of the 18th world computer congress.*, 2004.

[25] H. Zhang, A. Kankanhalli, and S. W. Smoliar. Automatic partitioning of full-motion video. *Multimedia Systems*, 1(1):10–28, 1993.

[26] Su Zhenglian. Improving the preformance of personalized recommendation with ontological user interest model. *Seventh International Conference on Computational Intelligence and Security*, 2011.