A Multi-Agent Open Architecture for a TV Recommender System: A Case Study using a Bayesian Strategy

MSE 2004 - December 15th, 2004

Alberto Gil Solla

agil@det.uvigo.es

Department of Telematic Engineering, University of Vigo (Spain)

Motivation of TV Recommender Systems

- Migration from analogue to digital TV.
- Implications:
  - More channels in the same bandwidth.
  - Software applications mixed with audiovisual contents.

Motivation of TV Recommender Systems

- Migration from analogue to digital TV.
- Implications:
  - More channels in the same bandwidth.
  - Software applications mixed with audiovisual contents.
- Disoriented users among large amount of irrelevant information.
  - User cannot use this new type of TV efficiently.
  - Necessary tools to find interesting TV programs

Related Work

- Different approaches in the field of TV personalization tools:
  - Bayesian techniques
  - Decision trees
  - Content-based methods
  - Collaborative filtering
  - ...

Related Work

- Different approaches in the field of TV personalization tools:
  - Bayesian techniques
  - Decision trees
  - Content-based methods
  - Collaborative filtering
  - ...
- A common base: **limitation in reasoning capabilities**.
  - Mechanisms to represent the knowledge of TV domain are not used in previous proposals.
  - Reasoning process allows to obtain enhaced recommendations.

AVATAR: Main Design Decisions (I)

- Two key elements in our system:
  - **TV-Anytime** specification for:
    - Generic descriptions of TV programs: title, genre, set of keywords, etc.
    - User preferences
    - User viewing history
  - Knowledge about TV domain provided by an **OWL ontology**.

AVATAR: Main Design Decisions (II)

- The system must be updated when user preferences change.
  - Goal: personalized and higher quality recommendations.

AVATAR: Main Design Decisions (II)

- The system must be updated when user preferences change.
  - Goal: personalized and higher quality recommendations.
- **AVATAR** must be flexible enough to favor updating process.
  - MHP applications tuned in user's receiver (Set-Top Box or STB).

AVATAR: Main Design Decisions (II)

- The system must be updated when user preferences change.
  - Goal: personalized and higher quality recommendations.
- **AVATAR** must be flexible enough to favor updating process.
  - MHP applications tuned in user's receiver (Set-Top Box or STB).
- MHP applications run in the context of a service or event.
  - Problem: All user actions must be recorded all the time: **local agent** to watch the viewer behaviour.
  - Local agent stores feedback information.
  - Normalized access by TV-Anytime MHP API.

AVATAR: The Architecture (I)

- A modular architecture based on multiple agents.
  - Efficient use of knowledge inference strategies.

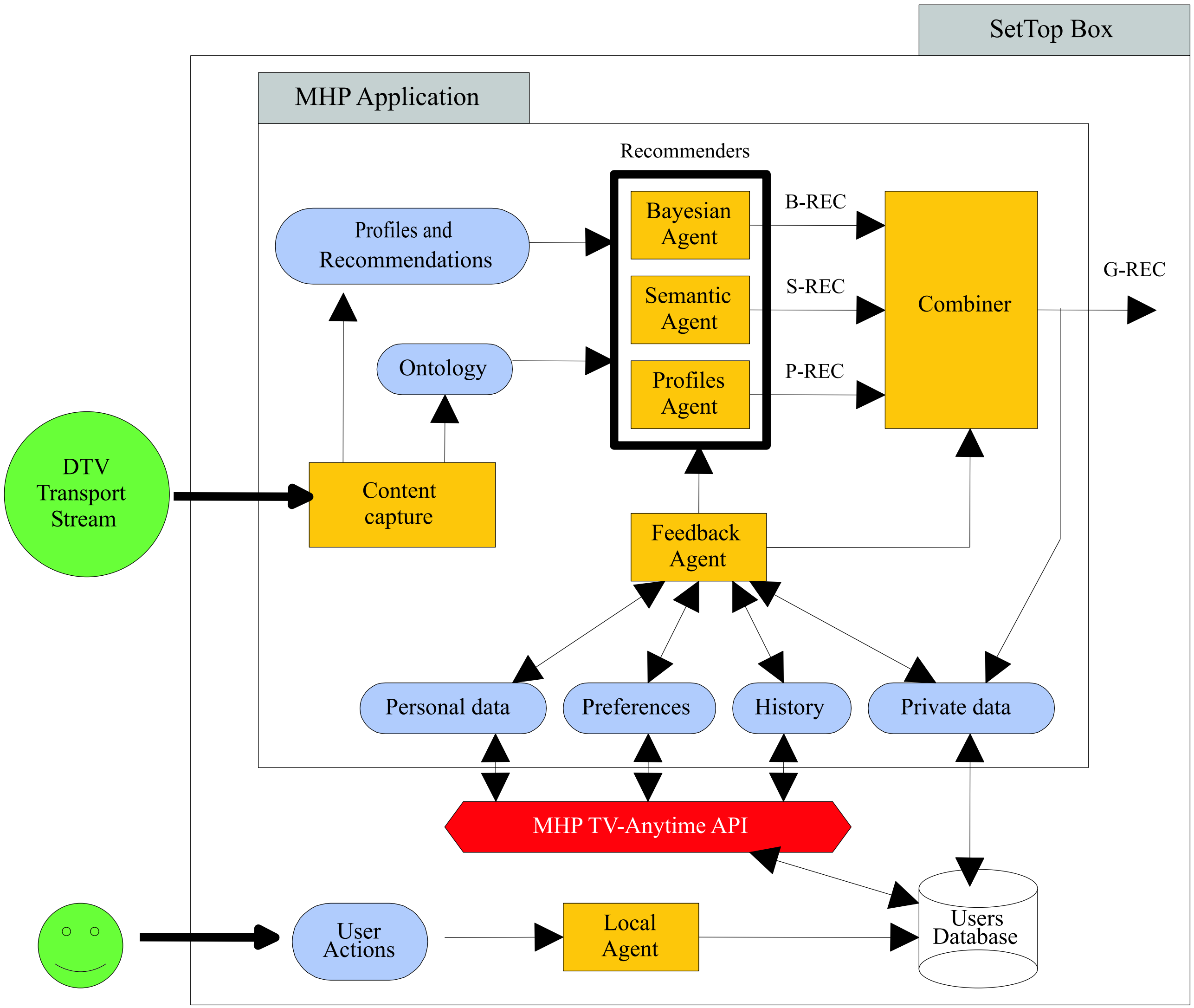
AVATAR: The Architecture (I)

- A modular architecture based on multiple agents.
  - Efficient use of knowledge inference strategies.
- Local software of user's STB
- MHP applications $\Rightarrow$ recommendation service

AVATAR: The Architecture (I)

- A modular architecture based on multiple agents.
  - Efficient use of knowledge inference strategies.
- Local software of user's STB
- MHP applications ⇒ recommendation service
  - Capture and classifications of received information
  - Recommenders:
    - **Naive Bayesian agents**
    - Agents based on profiles matching
    - Semantic agents
  - Feedback system

AVATAR: The Architecture (II)



The Naive Bayesian Classifiers

- Usefulness of Bayesian techniques in our approach
  - Codification of *uncertain* knowledge
  - Representation of *incomplete* knowledge
  - Implicit *collaborative recommendations*
  - Inference network easily updated

The Naive Bayesian Classifiers

- Usefulness of Bayesian techniques in our approach
  - Codification of *uncertain* knowledge
  - Representation of *incomplete* knowledge
  - Implicit *collaborative recommendations*
  - Inference network easily updated
- Input parameters are:
  - A class with a set of possible values
  - A group of input attributes
- The goal is to find out the most possible value of the input class, taking into account the values of the input attributes

The Naive Bayesian Classifiers (II)

- A tree is used as inference network
  - Root node $\Rightarrow$ The class to classify: V (v_j are the possible values)
  - Several childnodes $\Rightarrow$ The input attributes: $a_1, a_2, \ldots, a_n$

The Naive Bayesian Classifiers (II)

- A tree is used as inference network
  - Root node $\Rightarrow$ The class to classify: V (v_j are the possible values)
  - Several childnodes $\Rightarrow$ The input attributes: $a_1, a_2, \ldots, a_n$
- Mathematical expression to elabore recommendations

$$Rec(V) = argmax_{v_j \in V} = P(v_j)P(a_1, a_2, ..., a_n/v_j) \qquad (1)$$

- Supposition: Attributes conditionally independent once known the value v_j of V

$$P(a_1, ..., a_n/v_j) = \prod_i P(a_i/v_j) \qquad (2)$$

The Naive Bayesian Classifiers (III)

- Our approach uses *m-estimate* approximation to calculate P(a_i/v_j)

$$P(a_i/v_j) = \frac{n_c + mp}{n + m} \tag{3}$$

- Our system uses a large training set with information about users registered in AVATAR
  - n_c: number of examples where $v = v_j$ and $a = a_i$
  - n: number of examples where $v = v_j$
  - p: estimation of P(a_i/v_j)
  - m: *equivalent sample size* $\Rightarrow$ number of input attributes considered in our approach

Bayesian agents in AVATAR

- In our approach:
  - V are general categories of TV programs defined in our OWL ontology: e.g. *Movies*
  - v_j are more specific programs: e.g. *Action Movies*, *Drama Movies*, *Comedy Movies*
  - a_i are personal data about users and their preferences
- There exist one Bayesian agent for each category of TV programs $\Rightarrow$ A final recommendation is elaborated from the individual suggestions of each agent
- The training set is sent to the user's STB over the digital transport stream

Excerpt of the training set in AVATAR

Job	Likings	Marital Status	Viewed Series	Viewed Sports	Viewed Infor. Prog.	Viewed Movies	Recommended Movies
Engineer	Travel	Married	Comedies	Sporting News	News	Drama	**ACTION**
Doctor	Reading	Single	Drama	Sporting Events	Incidents	Drama	**DRAMA**
Doctor	Reading	Single	Action	Sporting Events	Incidents	Action	**COMEDIES**
Engineer	Travel	Single	Action	Sporting News	Incidents	Comedies	**COMEDIES**
Doctor	Travel	Married	Comedies	Sporting News	News	Action	**DRAMA**
Doctor	Travel	Married	Drama	Sporting News	News	Drama	**DRAMA**
Engineer	Travel	Single	Drama	Sporting Events	News	Action	**ACTION**
Engineer	Reading	Single	Drama	Sporting Events	News	Comedies	**ACTION**
Engineer	Reading	Married	Comedies	Sporting Events	Incidents	Comedies	**COMEDIES**
Doctor	Reading	Married	Comedies	Sporting Events	Incidents	Action	**COMEDIES**
Doctor	Travel	Married	Comedies	Sporting Events	News	Comedies	**DRAMA**
Engineer	Reading	Married	Action	Sporting News	News	Drama	**ACTION**
Engineer	Travel	Married	Action	Sporting News	News	Drama	**ACTION**
Doctor	Travel	Single	Action	Sporting News	News	Comedies	**COMEDIES**
Doctor	Reading	Single	Comedies	Sporting News	News	Action	**COMEDIES**
...	...	...	...	...	...	...	...

Example of Bayesian recommendation

- Necessary values to elaborate a recommendation of Movies:
    - V = Recommended Movies
    - $v_j \in$ {Action, Drama, Comedies}
    - $a_i \in$ {Job, Likings, Marital Status, Viewed Series, Viewed Sports, Viewed Inform. Programs, Viewed Movies}

Example of Bayesian recommendation

- Necessary values to elaborate a recommendation of Movies:
  - V = Recommended Movies
  - $v_j \in$ {Action, Drama, Comedies}
  - $a_i \in$ {Job, Likings, Marital Status, Viewed Series, Viewed Sports, Viewed Inform. Programs, Viewed Movies}
  - Assume a user with the following characteristics:
    - Job = Doctor
    - Likings = Travel
    - Marital Status = Single
    - Viewed Series = Action
    - Viewed Sports = Sporting News
    - Viewed Inform. Programs = News
    - Viewed Movies = Action

Example of Bayesian recommendation (II)

- Compute P(Recommended Movies = Comedy), P(Recommended Movies = Drama) and P(Recommended Movies = Action)
- Choose the maximum probability to recommend movies to the user
- Calculate P(a_i/v_j), with $v_j \in$ {Action, Drama, Comedies}
- For example, P(Doctor/Comedy) = $\frac{n_c+mp}{n+m} = \frac{4+\frac{7}{2}}{6+7} = \frac{15}{26}$
- In a similar way, P(Travel/Comedy), P(Single/Comedy), etc.

Example of Bayesian recommendation (II)

- Compute P(Recommended Movies = Comedy), P(Recommended Movies = Drama) and P(Recommended Movies = Action)
- Choose the maximum probability to recommend movies to the user
- Calculate P(a_i/v_j), with $v_j \in$ {Action, Drama, Comedies}
- For example, P(Doctor/Comedy) = $\frac{n_c+mp}{n+m} = \frac{4+\frac{7}{2}}{6+7} = \frac{15}{26}$
- In a similar way, P(Travel/Comedy), P(Single/Comedy), etc.
- V_{Action} = 0.0009289
- V_{Dramas} = 0.0008347
- $V_{Comedies}$ = 0.0020055 $\Rightarrow$ Bayesian agent recommends comedy movies

Conclusions and Further Work

- Open and modular architecture to favour an efficient use of recommendations methods
- Naive Bayesian techniques are useful in a recommendation context
  - They allow to handle uncertain and incomplete knowledge
  - Learning time is linear in the number of examples

Conclusions and Further Work

- Open and modular architecture to favour an efficient use of recommendations methods
- Naive Bayesian techniques are useful in a recommendation context
  - They allow to handle uncertain and incomplete knowledge
  - Learning time is linear in the number of examples
- Future work is related to the evaluation of more complex Bayesian techniques $\Rightarrow$ To avoid possible loss of precision in Naive classifiers
- We are working on the designing of semantic agents
  - Goal: To enhance the offered recommendations by semantic reasoning about descriptions of TV programs and user preferences

Thank you for your attention