QUALITY ENHANCEMENT BASED ON REINFORCEMENT LEARNING AND FEATURE WEIGHTING FOR A CRITIQUING-BASED RECOMMENDER

Maria Salamó, **Sergio Escalera**, and Petia Radeva

Computer Vision Center &
Universitat de Barcelona

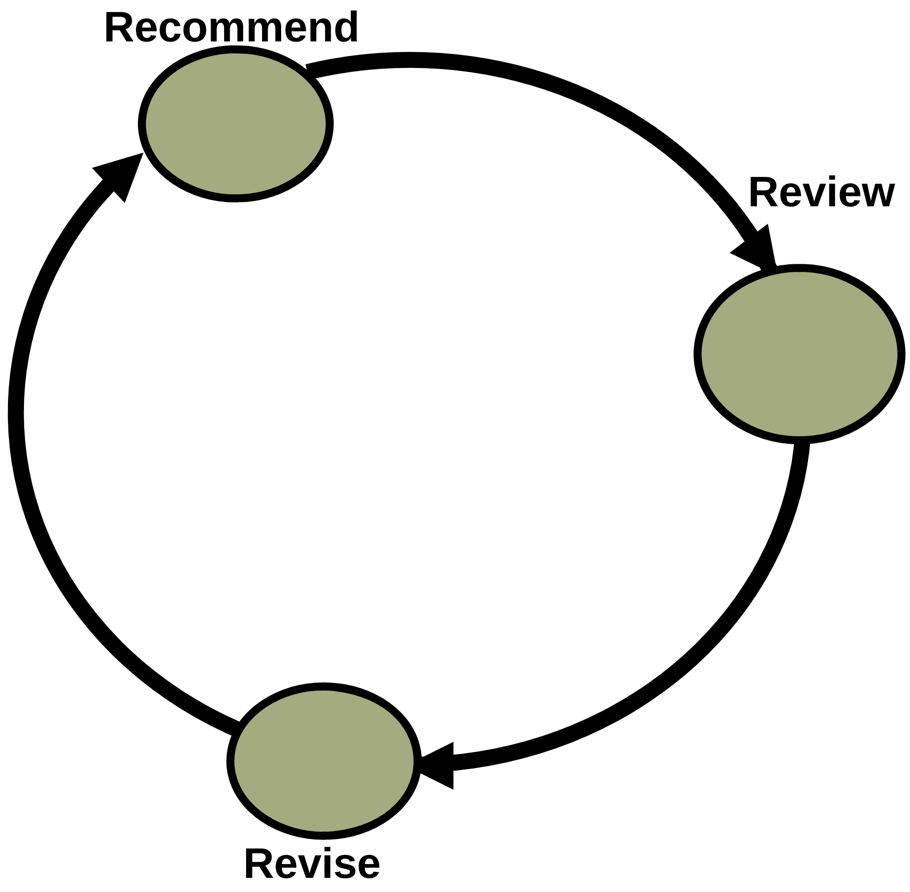
Outline

- Introduction
- Incremental Critiquing
- Proposals
- Compatibility using reinforcement learning
- Similarity using user preference weighting
- Results
- Conclusions

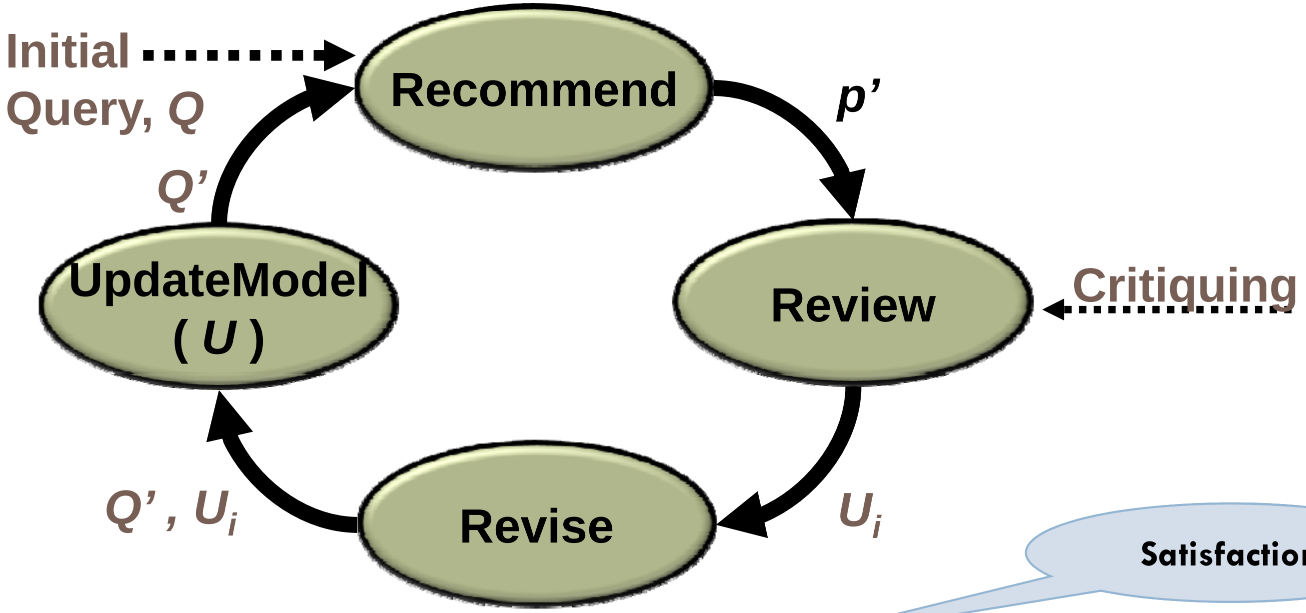
Introduction

Conversational recommenders play the role of an intelligent sales assistant guiding the user through a complex problem space by alternatively making **suggestions** and using **user feedback** to influence future suggestions.

The **feedback** in our recommender is based on **critiquing elicitation**



Incremental Critiquing



Satisfaction

$$C_t^{p'}(U) = \frac{\sum_{\forall_i:(1 \le i \le t)} \delta(p', U_i)}{|U|}$$

Compatibility

Similarity

Quality

$$Q(p', p, U) = \beta \cdot C_t^{p'}(U) + (1 - \beta) \cdot S(p', p)$$

Proposals

Compatibility

Similarity with weighting

$$Q(p', p, U) = \beta \cdot C_t^{p'}(U) + (1 - \beta) \cdot S(p', p)$$

- Different **reinforcement learning compatibility** functions
  - Monte-Carlo approaches
  - TD approaches
- Similarity using **user preference weighting**
  - Local user preference weighting [Salamó et al., 2005]
  - Global user preference weighting

The aim is to **enhance quality**, and thus, **reducing session length**

Compatibility using reinforcement learning

RL families:

- Dynamic Programming methods
  - Require a complete and accurate model of the environment
    - It is not possible define future behaviour of the user in the recommender
- Monte-Carlo methods
  - Do not require a model
- Temporal-Difference methods
  - Do not require a model

Compatibility using reinforcement learning

Both **Monte-Carlo** and **Temporal-Difference** methods seem to be useful to use the user experience

- Key Idea
  - Model the current compatibility of a candidate case *p'* at instant *t* based on its previous compatibility

Compatibility using reinforcement learning : **Monte-Carlo methods**

- **Monte-Carlo (MC)**

$$C_t^{p'} = C_{t-1}^{p'} + \alpha \cdot \left(R_t^{p'} - C_{t-1}^{p'} \right)$$

- **Exponential Monte-Carlo (MC)**

$$C_t^{p'} = \begin{cases} C_{t-1}^{p'} + \alpha \cdot \left(R_t^{p'} + C_{t-1}^{p'} \right) & \text{if } R_t^{p'} = 1 \\ C_{t-1}^{p'} - \alpha \cdot C_{t-1}^{p'} & \text{if } R_t^{p'} = 0 \end{cases}$$

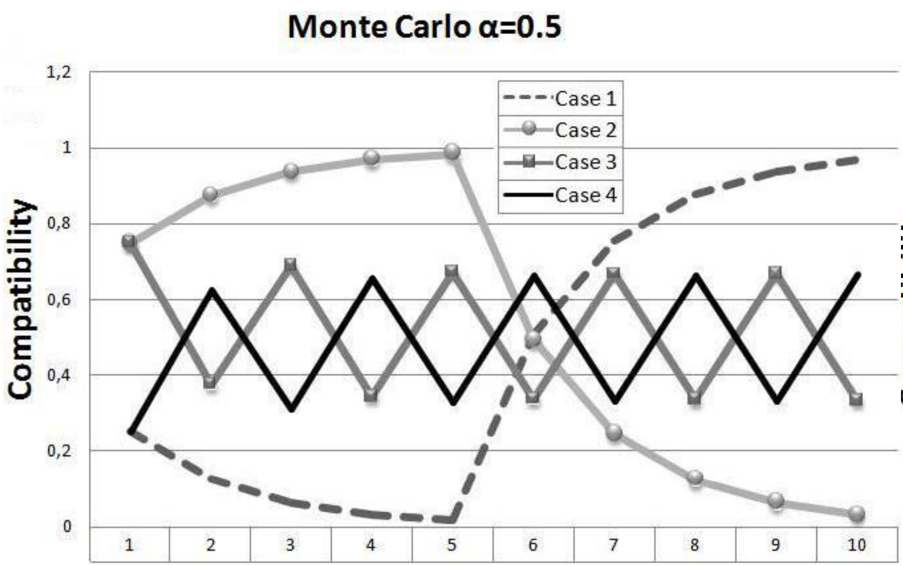
Compatibility using reinforcement learning : **Toy problem**

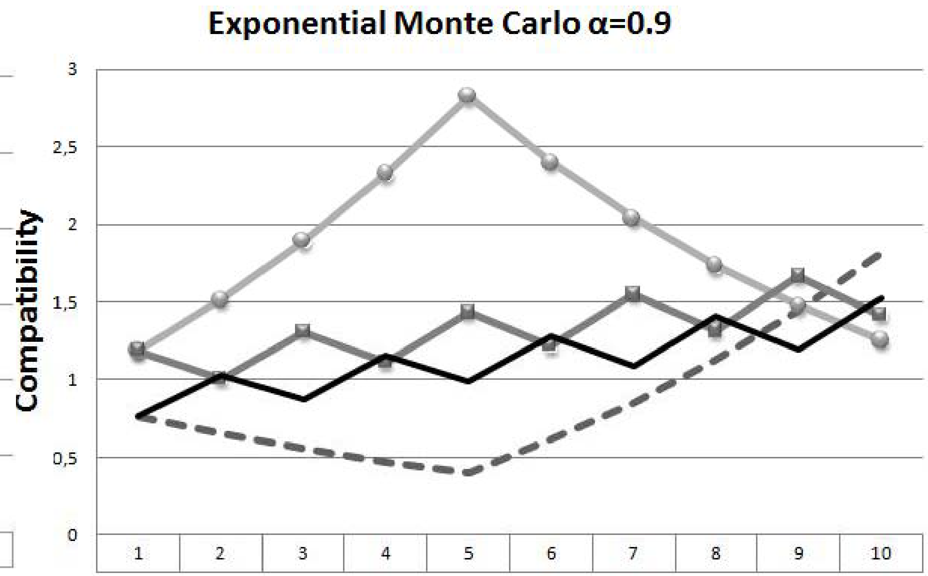
We use a toy problem to show the differences among strategies

- The toy problem contains:
  - Four cases
  - Ten cycles of the recommender
    - We suppose, for this example, that each cycle is an instant and each instant the recommender generates a critique (only one)
  - The critique satisfaction of each case at instant *t*
    - Satisfaction is 1 if the cases satisfies the critique, otherwise 0

	t=1	*t=2*	*t=3*	*t=4*	*t=5*	*t=6*	*t=7*	*t=8*	*t=9*	*t=10*
Case 1	0	0	0	0	0	1	1	1	1	1
Case 2	1	1	1	1	1	0	0	0	0	0
Case 3	1	0	1	0	1	0	1	0	1	0
Case 4	0	1	0	1	0	1	0	1	0	1

Compatibility using reinforcement learning : **MC and EMC comparison**





	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10
Case 1	0	0	0	0	0	1	1	1	1	1
Case 2	1	1	1	1	1	0	0	0	0	0
Case 3	1	0	1	0	1	0	1	0	1	0
Case 4	0	1	0	1	0	1	0	1	0	1

Compatibility using reinforcement learning : **Temporal-Difference methods**

- **Backward Temporal-Difference (BTD)**

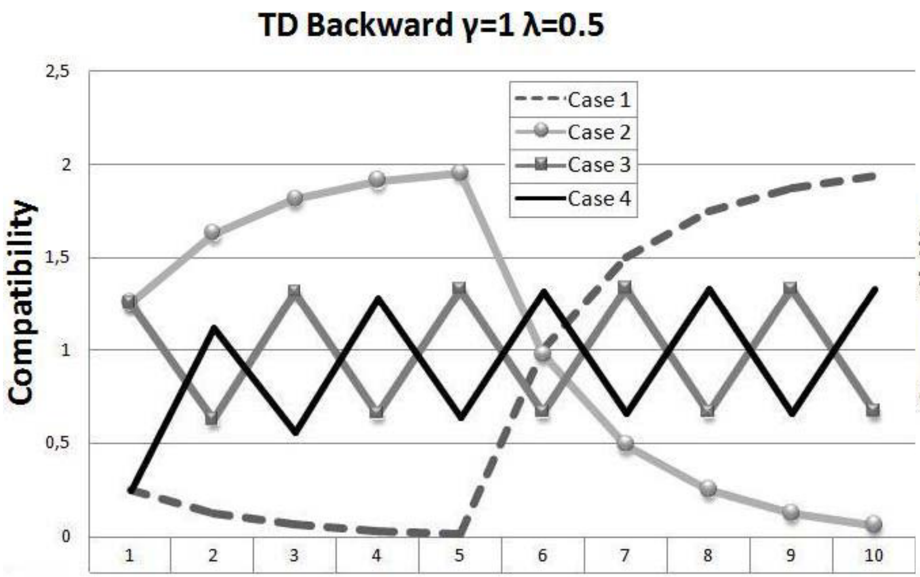
$$e_t^s = \begin{cases} \gamma \cdot \lambda \cdot e_{t-1}^s & \text{if } s \notin s_t \\ \gamma \cdot \lambda \cdot e_{t-1}^s + 1 & \text{if } s \in s_t \end{cases}$$

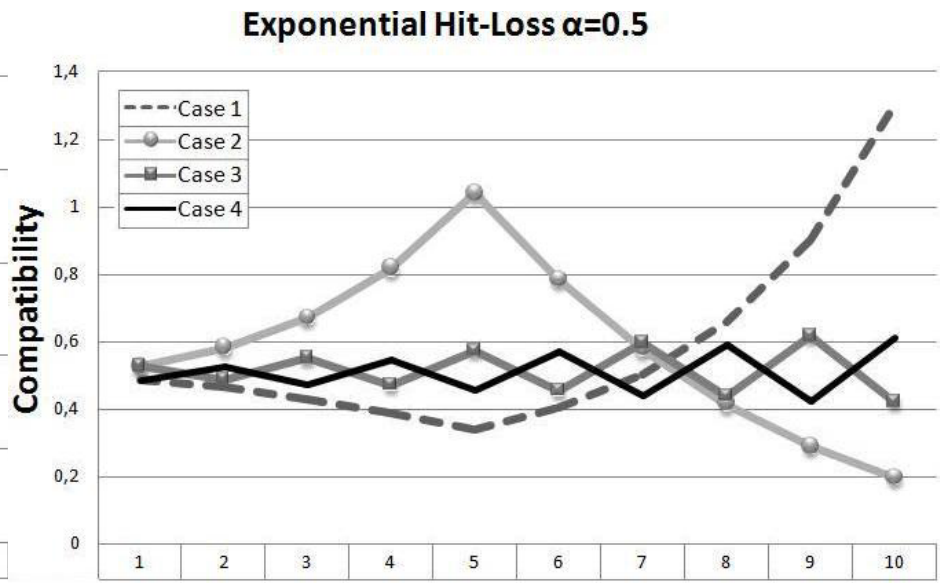
$$C_t^{p'} = \gamma \cdot \lambda \cdot C_{t-1}^{p'} + R_t^{p'}$$

- **Exponential Hit-Loss (EHL)**

$$C_t^{p'} = \begin{cases} h \leftarrow h+1, C_t^{p'} = C_{t-1}^{p'} \cdot (1+\alpha)^{(h^{p'}+t)k} & \text{if } R_t^{p'} = 1 \\ f \leftarrow f+1, C_t^{p'} = C_{t-1}^{p'} \cdot (1-\alpha)^{(f^{p'}+t)k} & \text{if } R_t^{p'} = 0 \end{cases}$$

Compatibility using reinforcement learning : **BTD and EHL comparison**





	t=1	t=2	t=3	t=4	t=5	t=6	t=7	t=8	t=9	t=10
Case 1	0	0	0	0	0	1	1	1	1	1
Case 2	1	1	1	1	1	0	0	0	0	0
Case 3	1	0	1	0	1	0	1	0	1	0
Case 4	0	1	0	1	0	1	0	1	0	1

Similarity using user preference weighting

- Similarity plays, as in traditional CBR, an important role in the recommender
  - As in CBR, similarity may improve by weighting features
- Key idea
  - To find the relative importance of each feature as a weighting value

Similarity Weight Distance

$$S(p',p) = \sum_{\forall f} W(p'_f) \cdot d(p'_f, p_f)$$

Similarity using user preference weighting: **Local user preference weighting (LW)**

- Key idea
  - Discovers the relative importance of each feature in each case as a weighting value
  - Prioritise those features that have not yet been critiqued

$$W(p'_f) = 1 - \frac{1}{2}\left(\frac{\sum_{\forall i \in U^f} \delta(p'_i, U^f_i)}{|U^f|}\right)$$

Similarity using user preference weighting: **Global user preference weighting (GW)**

- Key idea
  - Discovers a global vector of feature weights that will be used for the whole set of candidate cases
  - Prioritise those features that have not yet been critiqued

$$W(f) = 1 - \frac{1}{2}\left(\frac{\sum_{\forall p' \subseteq P'}^{\forall i \in U^f} \delta(p', U_i^f)}{|P'|}\right)$$

Results

Set-up

- Travel dataset which consists of 9 features and 1024 vacation cases
  - Contains numerical and nominal features
- We generate an artificial user that emulates the live users behaviour
- We analyse **easy, moderate** and **hard queries**
- *50* experiments repeated 10 times

Performance Criteria

- The average session length

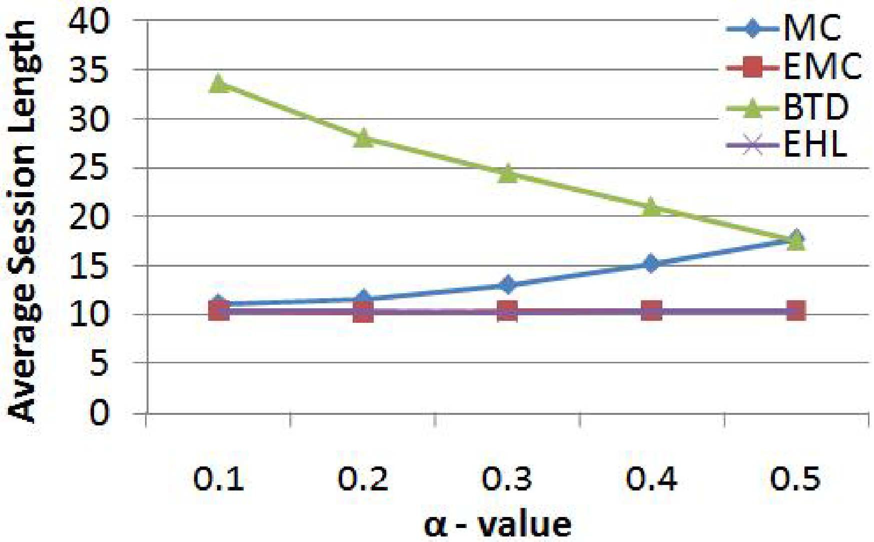
Statistics

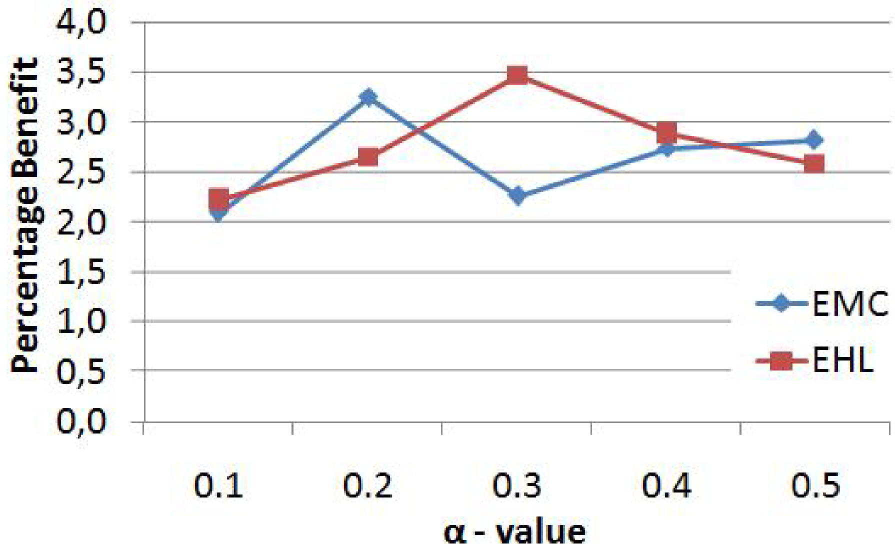
- Friedman test
- Nemenyi test

Results:

RL recommendation efficiency

Alpha analysis

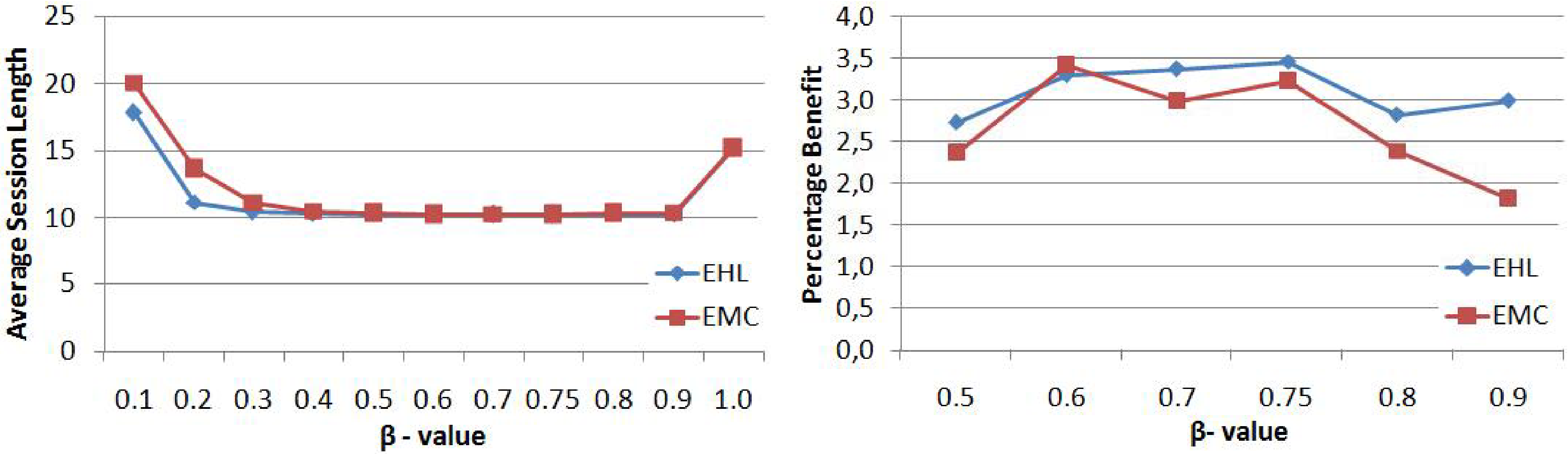




- MC and BTD present a tendency to increase/decrease the Avg. session length
- EMC and EHL (the ones who consider an exponential behaviour) results in shorter session length

Results:
RL recommendation efficiency

Beta analysis

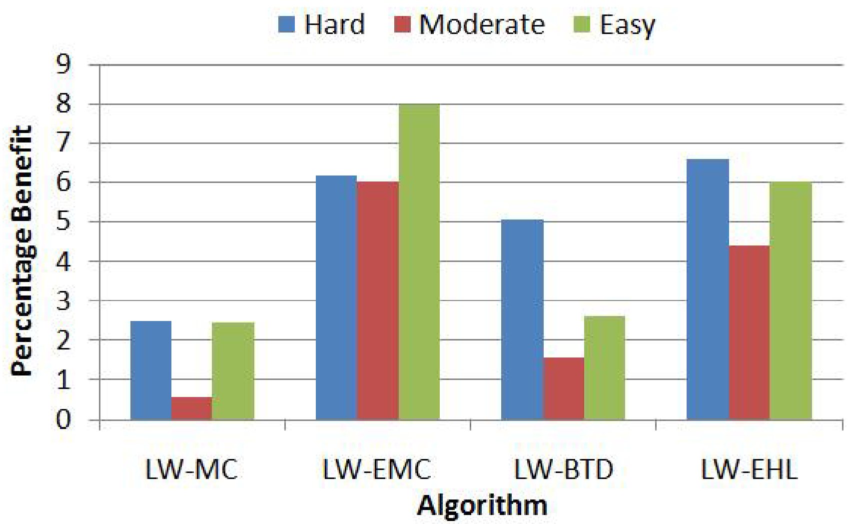


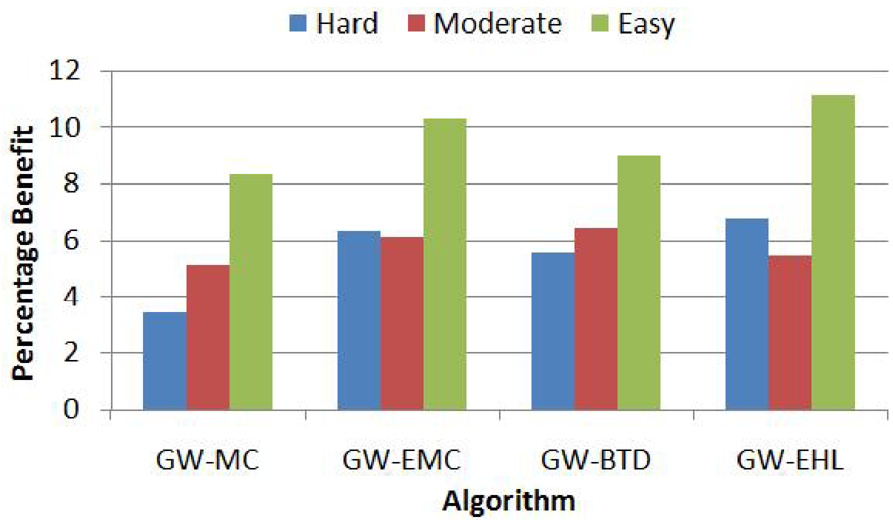
- Session lengths are maintained between 0.5 to 0.9
- Best results are for 0.6 and 0.75
- We set up this value for our next experiments

Results:

Quality Recommendation efficiency

Comparison of LW and GW with RL measures



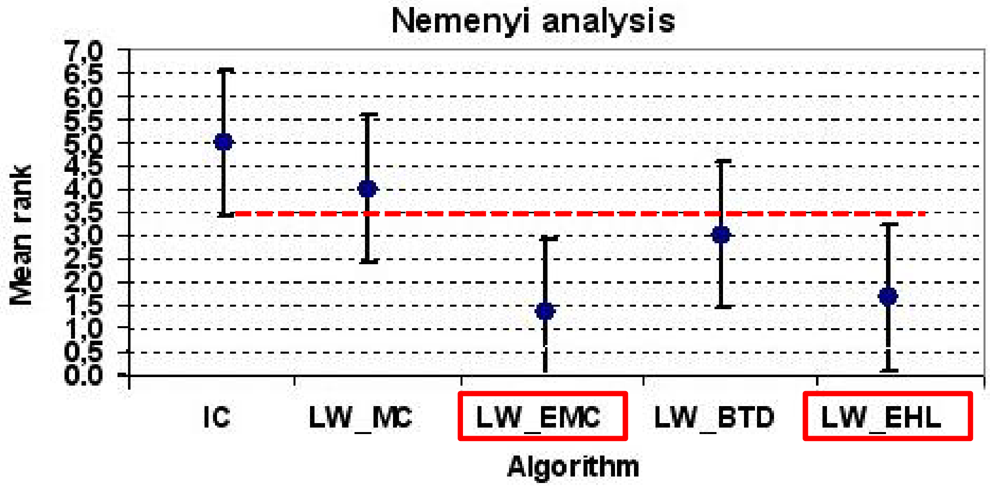


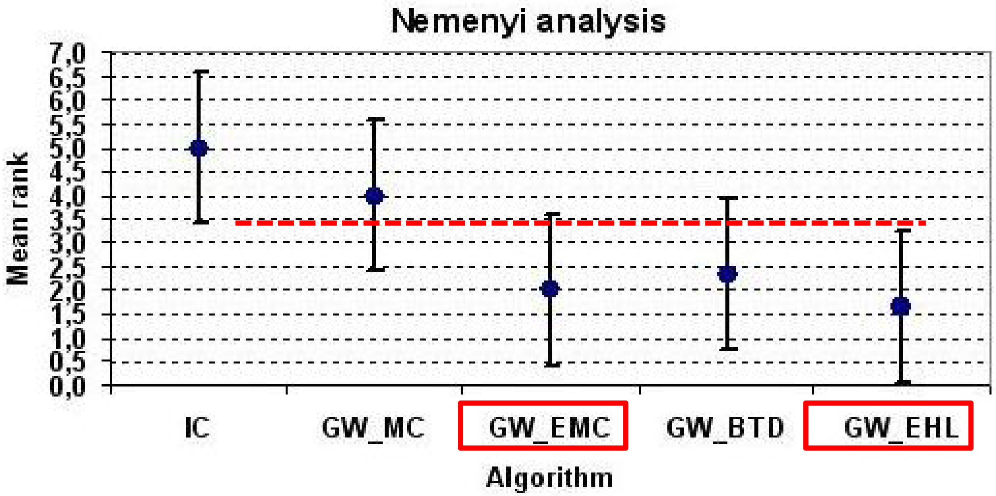
- The combinations of LW with RL measures result in a reduction in session length that ranges from 0,5% up to 8%
- GW combinations with RL measures present the highest benefit, ranging from 3,4% up to 11,1%

Results:

Quality recommendation efficiency

- **Friedman test**
  - Five algorithms
  - Three different queries
  - $F(4,8) = 3.83$ at the 0.05 critical level
  - $F_F = 40.06$ (LW)
  - $F_F = 9.22$ (GW)
  - We can reject the null hypothesis in both analysis
- Nemenyi test
  - Critical difference is 3.17





Conclusions & future work

- We have proposed new strategies for compatibility computation and feature weighting that enhance quality
- The new compatibility strategies offer better benefit in terms of session length
- Global user preference weighting shows significant improvements in comparison to the state-of-the-art-approaches

- More data to test: Influence of dimensionality?
- Real user evaluation
- Current work: introducing recommendation to retrieve cases from audio and video data sets

Thank you for your attention

Maria Salamó, Sergio Escalera, and Petia Radeva

Computer Vision Center &
Universitat de Barcelona