Part 15: Context Dependent Recommendations

Francesco Ricci
Free University of Bozen-Bolzano
Italy
fricci@unibz.it

Content

- What is context?
- Types of context
- Context impact on recommendations and ratings
- Context modelling – collaborative filtering
- Context-based recommendation computation
- When context matters – detecting relevance
- Application: InCarMusic
- Contextual computing
- Adapting the recommendation to the current interaction context.

Motivating Examples

- Recommend a vacation
  - **Winter** vs. **summer**
- Recommend a purchase
  - **Gift** vs. for **yourself**
- Recommend a movie
  - With **girlfriend** in a **movie theater** vs. at **home** with a **group of friends**
- Recommend a recipe
  - **Alone** vs. with **my kids**
- Recommend music
  - When you have a **happy** vs. **sad mood**.

These contextual factors can change the evaluation/rating of the user for the considered item – and the user's choices

Context in Recommender Systems

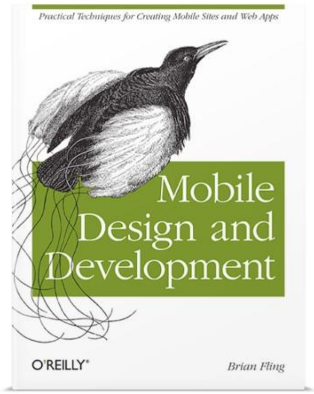
- *Recommender Systems are software tools and techniques providing suggestions for items to be of use to a user*

Context is any information or conditions that can influence the *perception* of the usefulness of an item for a user

- Recommender systems must take into account this information to deliver more **useful** (perceived) recommendations.

[Adomavicius and Tuzhilin, 2011]

Types of Context - Mobile



[Fling, 2009]

- **Physical context**
  - **time**, **position**, and **activity** of the user, weather, light, and temperature ...
- **Social context**
  - the presence and role of other people around the user
- **Interaction media context**
  - the device used to access the system and the type of media that are browsed and personalized (text, music, images, movies, …)
- **Modal context**
  - The state of mind of the user, the user’s goals, mood, experience, and cognitive capabilities.

Example: Factors influencing Holiday Decision

Internal to the tourist

Personal Motivators

Personality

Disposable Income

Health

Family commitments

Past experience

Works commitments

Hobbies and interests

Knowledge of potential holidays

Lifestyle

Attitudes, opinions and perceptions

Decision

External to the tourist

Availability of products

Advice of travel agents

Information obtained from tourism organization and media

Word-of-mouth recommendations

Political restrictions: visa, terrorism,

Health problems

Special promotion and offers

Climate

[Swarbrooke & Horner, 2006]

Ratings in Context

- Rating: measures how much a user likes an item – general definition – *without substance*
- We believe that it is linked to the goodness of a recommendation:
  - *The larger the rating the higher is the probability that the recommended item suits to the user*
- **Not always:**
  - I like Ferrari cars (5 stars) but it is unlikely that I will buy one!
  - I gave 5 stars to a camera – this does not mean that I will buy another camera if I have one
- Only **in context** we can transform a rating into a measure of the likelihood to choose an item (utility)

Examples: Music Recommendation

- *I like Shoenberg string trio op. 45 but it is unlikely that I will play it on Christmas Eve*
- *I'm fond of Stravinsky chamber music but after 2 hours of repeated listening to such music I like something different*
- *When approaching the Bolzano gothic cathedral I find more appropriate to listen to Bach than to U2*
- *When traveling by car with my family I typically listen to pop music that I otherwise "hate"*
- *When traveling along the coastline I will enjoy listening to Blues music.*

What Context is Relevant?

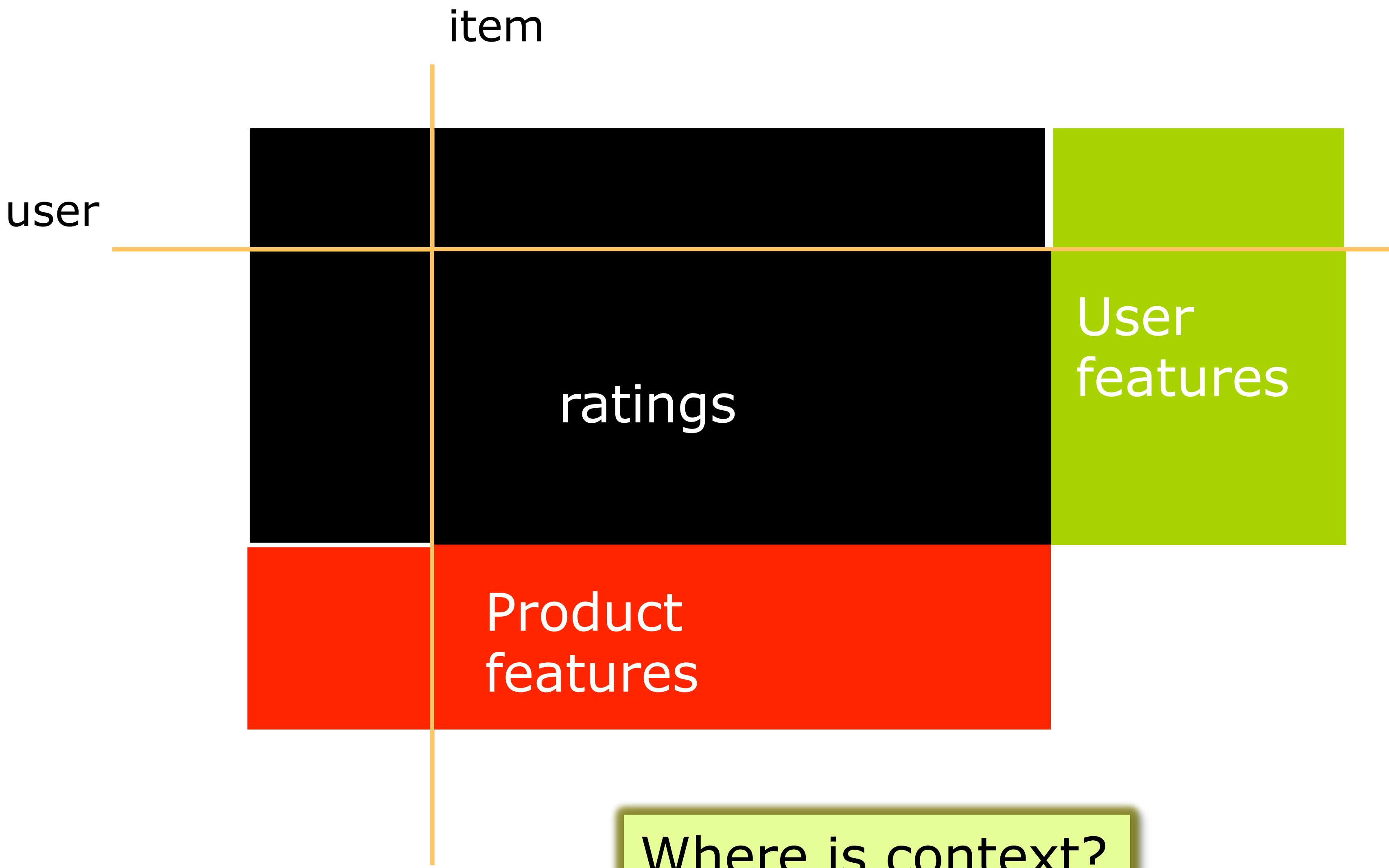
- "Shindler's List" has been *rated* 5 stars by john on January 27th (Remembrance day)
  - *In this case January 27th is expressing relevant context*
- "Shindler's List" has been *rated* 4 stars by john on March 27th
  - *In this case March 27th is expressing (probably) irrelevant context*
- **Context relevance may be item dependent**
- … and also user dependent
- **What are the relevant contextual dimensions and conditions for each item and user?**

A Traditional (Bi-dimensional) Model of Recommendation

1. Two types of entities: **Users** and **Items**
2. A **background knowledge**:
   - A set of ratings: a map R: Users x Items $\twoheadrightarrow$ [0,1] U {?} **– R is a partial function!**
   - A set of “features” of the Users and/or Items
3. A **method** for **substituting** all or part of the ‘?’ values - for some (user, item) pairs – with good rating predictions
4. A method for **selecting the items to recommend**
   - Recommend to u the item:
   - $i^* = \arg\max_{i \in Items} \{R(u,i)\}$

[Adomavicius et al., 2005]

A Bidimensional Model



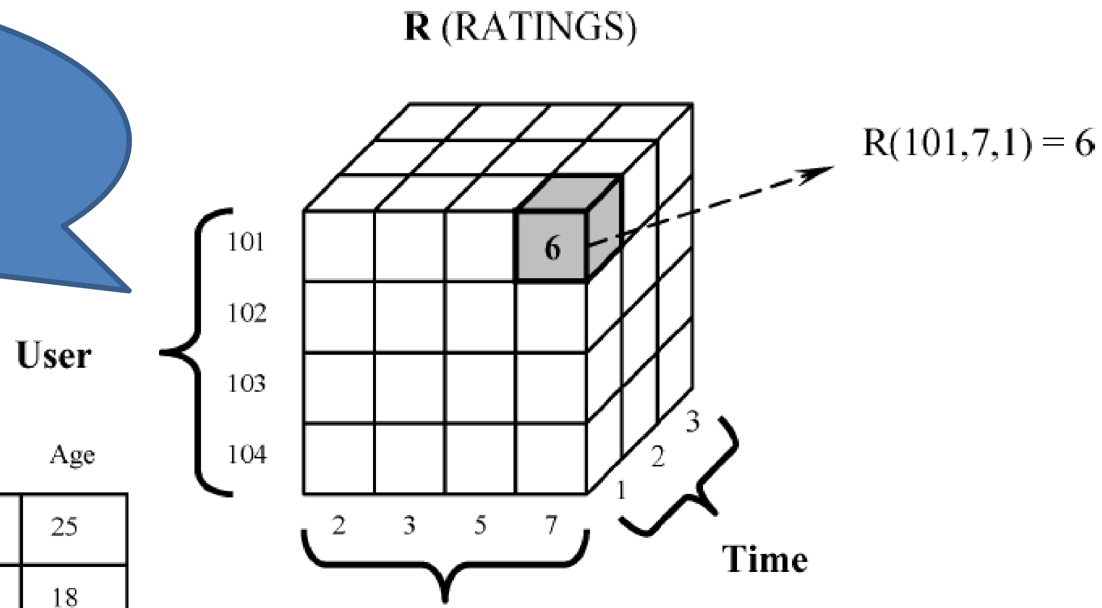
Where is context?

Bi-dimensional vs. multidimensional

- The previous model (R: Users x Items $\twoheadrightarrow$[0,1] U {?}) is **bi-dimensional**
- A more general model may include "**contextual**" dimensions, e.g.:
  - R: **Users** x **Time** x **Goal** x **Items** $\twoheadrightarrow$[0,1] U {?}
- **Assumption:** the rating function or, more in general, the recommendation evaluation is more complex than an assignment of each pair (user, product) to a rating
- There must be some "hidden variables" that contributes to determining the rating function
- This multidimensional data model approach was developed for data warehousing and OLAP.

Multidimensional Model

Pozor – například pro lokaci uživatele je obvykle třeba použít jiný přístup (multicriterial optimization)



User

Id	Name	Age
101	John	25
102	Bob	18
103	Alice	27
104	Mary	21

Item

Id	Name	Cost
2	AB17	250.00
3	AB23	299.95
5	XY70	150.00
7	ZZ55	115.50

Time

Id	Name
1	Weekday
2	Weekend
3	Holiday

[Adomavicius et al., 2005]

General Model

- D_1, D_2, … D_n are **dimensions**
- The **recommendation space** is n-dimensional: $D_1 \times D_2 \times \ldots \times D_n$
- Each dimension is a subset of the Cartesian product of some attributes $D_i \subseteq A_{i(1)} \times \ldots \times A_{i(ki)}$ – **profile** of the dimension D_i

- **General Rating function**
  - $R: D_1 \times D_2 \times \ldots \times D_n \twoheadrightarrow [0,1] \cup \{?\}$

Problémem je ale data sparsity

[Adomavicius et al., 2005]

Recommendation Problem

- Assume that the rating function is **complete** (defined for each entry in $D_1 \times D_2 \times \ldots \times D_n$)
- Recommendation problem:
  - **"what"** to recommend is a subset of the dimensions: $D_{i1}, \ldots, D_{ik}$ $(k<n)$
  - **"for whom"** is another subset of the dimensions: $D_{j1}, \ldots, D_{jl}$ $(l<n)$
  - The dimension in "what" and "for whom" have a void intersection, and

for whom

$$\forall (d_{j1}, \ldots, d_{jl}) \in D_{j1} \times \cdots \times D_{jl}, \quad (d_{i1}, \ldots, d_{ik}) = \underset{\substack{(d'_{i1}, \ldots, d'_{ik}) \in D_{i1} \times \cdots \times D_{ik} \\ (d'_{j1}, \ldots, d'_{jl}) = (d_{j1}, \ldots, d_{jl})}}{\arg\max} R(d'_1, \ldots, d'_n)$$

what

This is given

Example

- **Movie:** defined by attributes *Movie(MovieID, Name, Studio, Director, Year, Genre, MainActors)*
- **Person:** defined by attributes *Person(UserID, Name, Address, Age, Occupation, etc.)*
- **Place:** a single attribute defining the listing of movie theaters and also the choices of the home TV, VCR, and DVD
- **Time:** the time when the movie can be or has been seen: *Time(TimeOfDay, DayOfWeek, Month, Year)*
- **Companion:** a person or a group with whom one can see the movie: a single attribute having values "alone," "friends," "girlfriend/boyfriend," "family," "co-workers," and "others."

Example (cont)

`R(movie, person, place, time, companion)` context

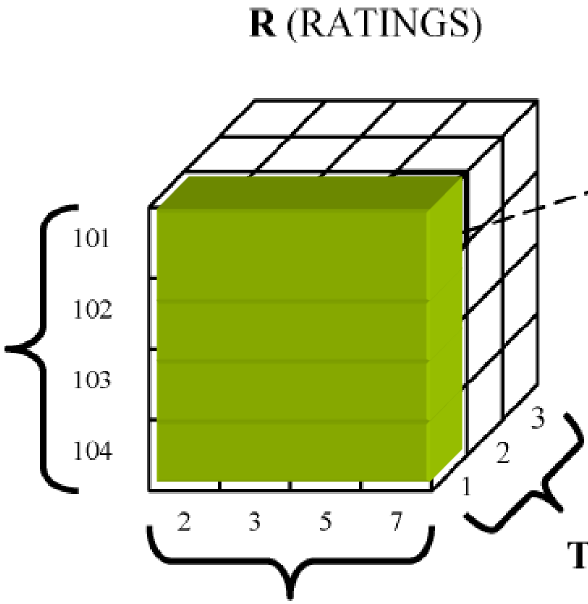
- Recommend the best movies to users
- Recommend top 5 action movies to users older than 18
- Recommend top 5 movies to user to see on the weekend, but only if the personal ratings of the movies are higher than 0.7
- Recommend to Tom and his girlfriend top 3 movies and the best time to see them over the weekend
- Recommend movie genre to different professions using only the movies with personal ratings bigger than 6.

Reduction-Based (pre-filtering)

- 1) Reduce the problem of multimensional recommendation to the traditional two-dimensional User x Item
- *2) For each "value" of the contextual dimension(s) estimate the missing ratings with a traditional method*
- Example:
  - R: U x I x T →[0,1] U {?} ; User, Item, Time
  - $R^{D}(u, i, t) = R^{D[T=t]}(u, i)$ Estimation based on data D, such that T=t
  - *The context-dependent estimation for (u, i, t) is computed using a traditional approach, in a two-dimensional setting, but using only the ratings that have T=t.*

Multidimensional Model

We use only the slice for T=t



User

Id	Name	Age
101	John	25
102	Bob	18
103	Alice	27
104	Mary	21

Item

Id	Name	Cost
2	AB17	250.00
3	AB23	299.95
5	XY70	150.00
7	ZZ55	115.50

Time

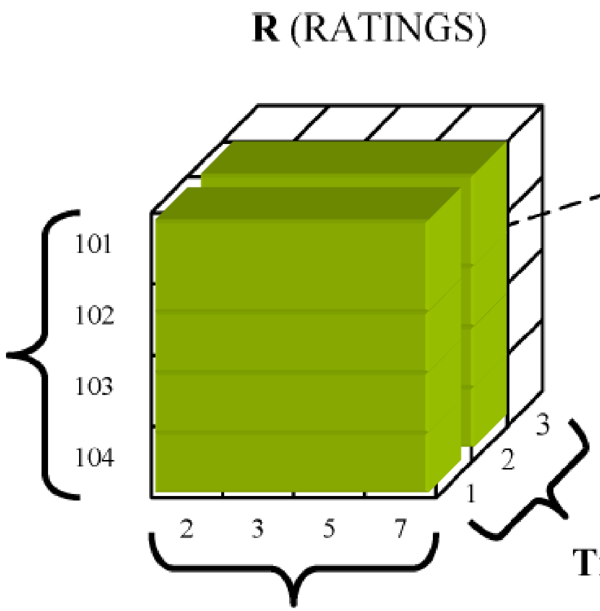
Id	Name
1	Weekday
2	Weekend
3	Holiday

Problems with the reduction

- The relation *D[Time=t](User, Item, Rating)* may not contain enough ratings for the two dimensional recommender algorithm to accurately predict *R(u, i)* for that specific value *t* of the Time variable
- **Approach:** use a "larger" contextual segment S_t, such that $t \in S_t$
- Instead of $R^D(u,i,t) = R^{D[T=t]}(u,i)$
- We have $R^D(u,i,t) = R^{D[t \in St]}(u,i)$ **aggregated**
- **Example:** instead of considering only the ratings of a specific day, e.g., Monday, use the ratings of all the weekdays and aggregate them to produce a two-dimensional slice.

Multidimensional Model

We use the slices for T=t, and T=t’ and we merge the two slices with an aggregation function, e.g., AVG



User

Id	Name	Age
101	John	25
102	Bob	18
103	Alice	27
104	Mary	21

Item

Id	Name	Cost
2	AB17	250.00
3	AB23	299.95
5	XY70	150.00
7	ZZ55	115.50

Time

Id	Name
1	Weekday
2	Weekend
3	Holiday

Research Problem

- **Local vs. Global model:** the local model exploits the local context "around" a particular user-item interaction to build the prediction, whereas the global model of CF uses all the user-item interactions - ignoring the contextual information
- Will a local model always outperform the global model?
- Is the local variability worth exploiting?
- When there is a “dependency” between context and rating?
- When the contextual dimensions will not reduce the available data to a too tiny subset?

Algorithms and Performance

- $\mu_{A,S}(S)$ is a (cross validated) measure of performance computed using only the ratings in the segment (contextual dependent)
- $\mu_{A,T}(S)$ is the same (cross validated) measure of performance but computed using all the data
- To compute both $\mu_{A,S}(S)$ and $\mu_{A,T}(S)$ they use: **user to user collaborative filtering**
- They have used as measure of performance **F1**

Finding high-performance segments

Inputs:

T set of pre-specified ratings for a multidimensional recommendation space.

$R_{A,T}$ rating estimation function based on algorithm A and training data T.

μ performance metric function.

N threshold defining the minimal number of ratings for a "large" segment.

Outputs:

$SEGM(T)$ – set of contextual segments on which the reduction-based approach based on algorithm A significantly outperforms the pure algorithm A.

Algorithm:

1. Let $SEGM(T)$ initially be the set of all large contextual segments for the set of ratings T.
2. For each segment $S \in SEGM(T)$ compute $\mu_{A,S}(S)$ and $\mu_{A,T}(S)$, and keep only those segments $S \in SEGM(T)$ for which $\mu_{A,S}(S)$ is *better*[5] than $\mu_{A,T}(S)$.
3. Among the segments remaining in $SEGM(T)$ after Step 2, discard any segment S for which there exists a different segment Q such that $S \subset Q$ and $\mu_{A,Q}(Q)$ is better than $\mu_{A,S}(S)$. The remaining segments form $SEGM(T)$.

Segments where context-awareness pays off

Finding the "Large" segments

- A segment is a "logical" aggregation of ratings based on some contextual dimensions: e.g., the ratings collected in the "week end", or the ratings in the "week end at home"
- Not easy to find all large segments with enough data
- Classical clustering/partitioning problem
- Rely on background information (such as those provided by a marketing expert) to determine the initial segments
- Use the "natural" hierarchies on the contextual dimensions to determine the segments.

Combining the local and global predictions

- Basic idea of the **combined approach** here proposed for context exploitation:
  1. Local: Use the prediction of the **best performing segments** to which a point belongs
  2. Global: If there is **no segments that contain the point use the standard prediction**, that is, computed without using any segment
- Hence the combined approach **will always work better or equal than the standard approach** (at the cost of the additional search on the set of segments)
- *BUT: how much better? Is it worth the extra effort?*

Combining the local and global predictions

The larger the performance value the better the segment

Inputs:

$SEGM(T)=\{S_1, \ldots, S_k\}$—where segments S_1 through S_k are arranged in the decreasing order with respect to μ, i.e., $\mu_{A,S_1}(S_1) \geq \cdots \geq \mu_{A,S_k}(S_k)$.

d—data point for which we want to estimate the rating.

Outputs:

$d.R$—estimated rating for data point d.

Algorithm:

$done$ = `false`; $i = 1$;
while $(i \leq k)$ **and** $(\neg done)$ **do** {
  if $d \in S_i$ **then** { $d.R = R_{A,Sj}(d)$;
      $done$ = `true` }
  $i = i + 1$ }
if $(\neg done)$ **then** $d.R = R_{A,T}(d)$ // i.e., d does not belong to any segment S_i

Prediction based on algorithm *A* and data S_j

Experimental Evaluation

- Acquired movie ratings and contextual information related to
  - **Time:** weekday, weekend, don’t remember
  - **Place:** movie theater, at home, don’t remember
  - **Companion:** alone, with friends, with partner, with family, others
- Movies rated in a scale **from 1 to 13**
- Participants were **students**
- 1755 ratings by 117 students over a period of 12 months
- Dropped students that had rated less than 10 movies
- **Finally 62 students, 202 movies and 1457 ratings (the set T) – not very big!**

Searching large segments

Name	Size	Description
Home	727	Movies watched at home
Friends	565	Movies watched with friends
NonRelease	551	Movies watched not during the 1st weekend of their release
Weekend	538	Movies watched on weekends
Theater	526	Movies watched in the movie theater
Weekday	340	Movies watched on weekdays
GBFriend	319	Movies watched with girlfriend/boyfriend
Theater-Weekend	301	Movies watched in the movie theater on weekends
Theater-Friends	274	Movies watched in the movie theater with friends

- These are obtained by performing an exhaustive search among the space of all possible segments (for the different dimensions try all different attribute values combinations)
- Each one of these segments has more than 262 user-specified ratings *(more than 20% of the dataset D_M – the training data set used for finding the segments – 90% of T)*

Comparison on each segment

p=0.025
z= -1.96

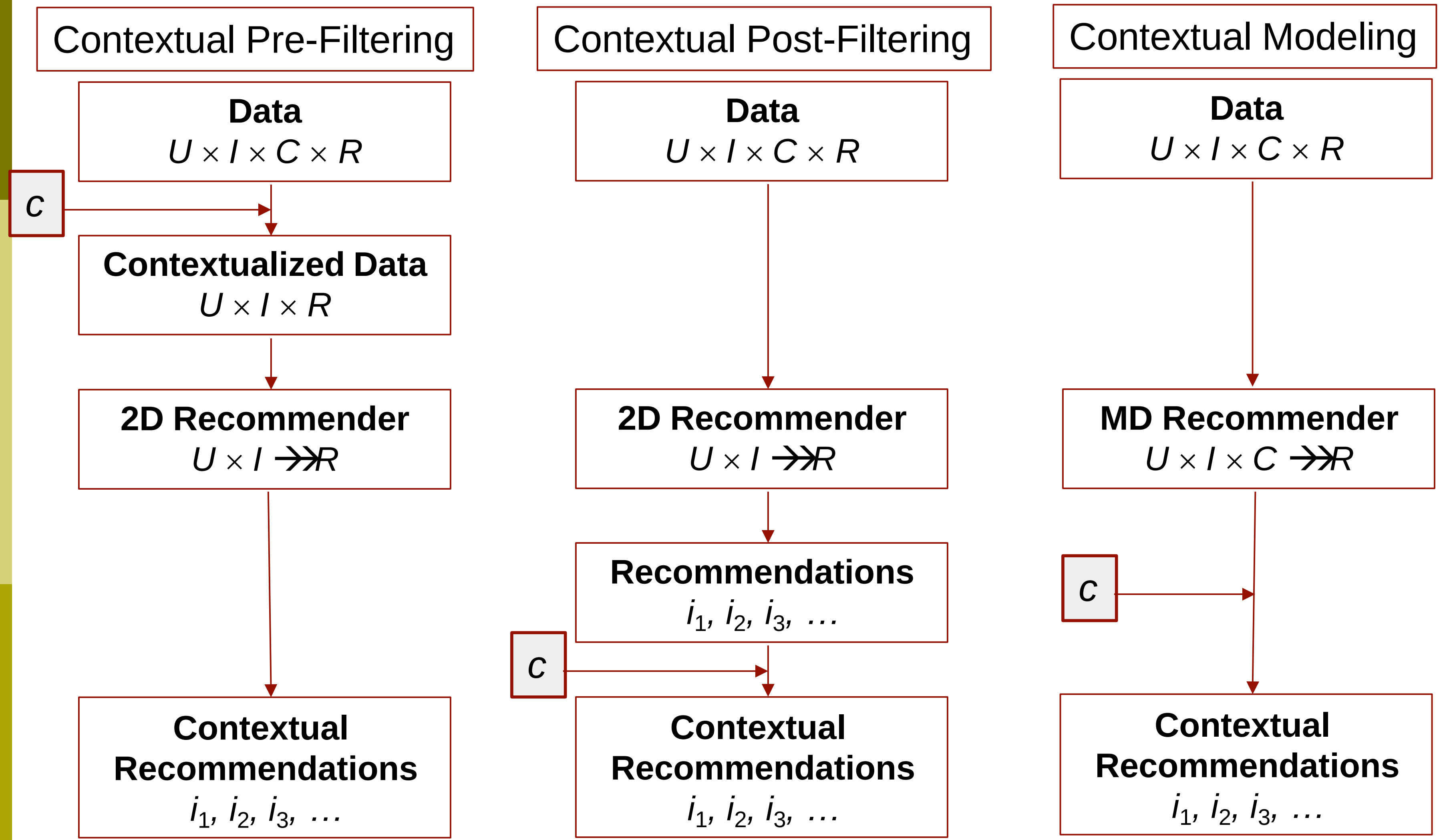
Segment	Method (CF)	Precision	Recall	F-measure
Home	Segment-based	0.527	0.319	0.397
Segment size: 727	Whole-data-based	0.556	0.357	0.435
Predicted: 658	*z-values*	*0.427*	*0.776*	
Friends	Segment-based	0.526	**0.444**	0.482
Segment size: 565	Whole-data-based	0.643	0.333	0.439
Predicted: 467	*z-values*	*1.710*	***−2.051***	
NonRelease	Segment-based	0.495	0.383	0.432
Segment size: 551	Whole-data-based	0.500	0.333	0.400
Predicted: 483	*z-values*	*0.065*	*−0.869*	
Weekend	Segment-based	0.596	**0.497**	**0.542***
Segment size: 538	Whole-data-based	0.655	0.383	0.484
Predicted: 463	*z-values*	*0.983*	***−2.256***	
Theater	Segment-based	0.622	**0.595**	**0.608***
Segment size: 526	Whole-data-based	0.694	0.366	0.479
Predicted: 451	*z-values*	*1.258*	***−4.646***	
Weekday	Segment-based	0.415	0.349	0.379
Segment size: 340	Whole-data-based	0.531	0.270	0.358
Predicted: 247	*z-values*	*1.041*	*−0.964*	
GBFriend	Segment-based	0.513	0.451	0.480
Segment size: 319	Whole-data-based	0.627	0.352	0.451
Predicted: 233	*z-values*	*1.292*	*−1.361*	
Theater-Weekend	Segment-based	0.660	**0.623**	**0.641***
Segment size: 301	Whole-data-based	0.754	0.406	0.528
Predicted: 205	*z-values*	*1.234*	***−3.161***	
Theater-Friends	Segment-based	0.657	**0.564**	**0.607***
Segment size: 274	Whole-data-based	0.732	0.385	0.504
Predicted: 150	*z-values*	*0.814*	***−2.245***	

Summary of the differences

Segment	Segment-based F-measure	Whole-data-based F-measure
Theater-Weekend	0.641	0.528
Theater	0.608	0.479
Theater-Friends	0.607	0.504
Weekend	0.542	0.484

- Substantial improvement of F-measure on some segments
- Since Theater-Friends has lower F-measure than Theater then this is discarded (see the original algorithm)
- The final segments obtained are: Theater-Weekend, Theater and Weekend.

Paradigms for Incorporating Context in Recommender Systems



[Adomavicius and Tuzhilin 2008]

How to detect context relevancy

- It is unrealistic to believe that one can detect the relevance of context by mining the data
  - Think about the detection of the importance of "January 27th" for "Shindler's List" – you will never discover that
- It is impossible to avoid the usage of explicit knowledge – before using data mining techniques

Data mining can refine reasonably defined hypothesis

Methodological Approach

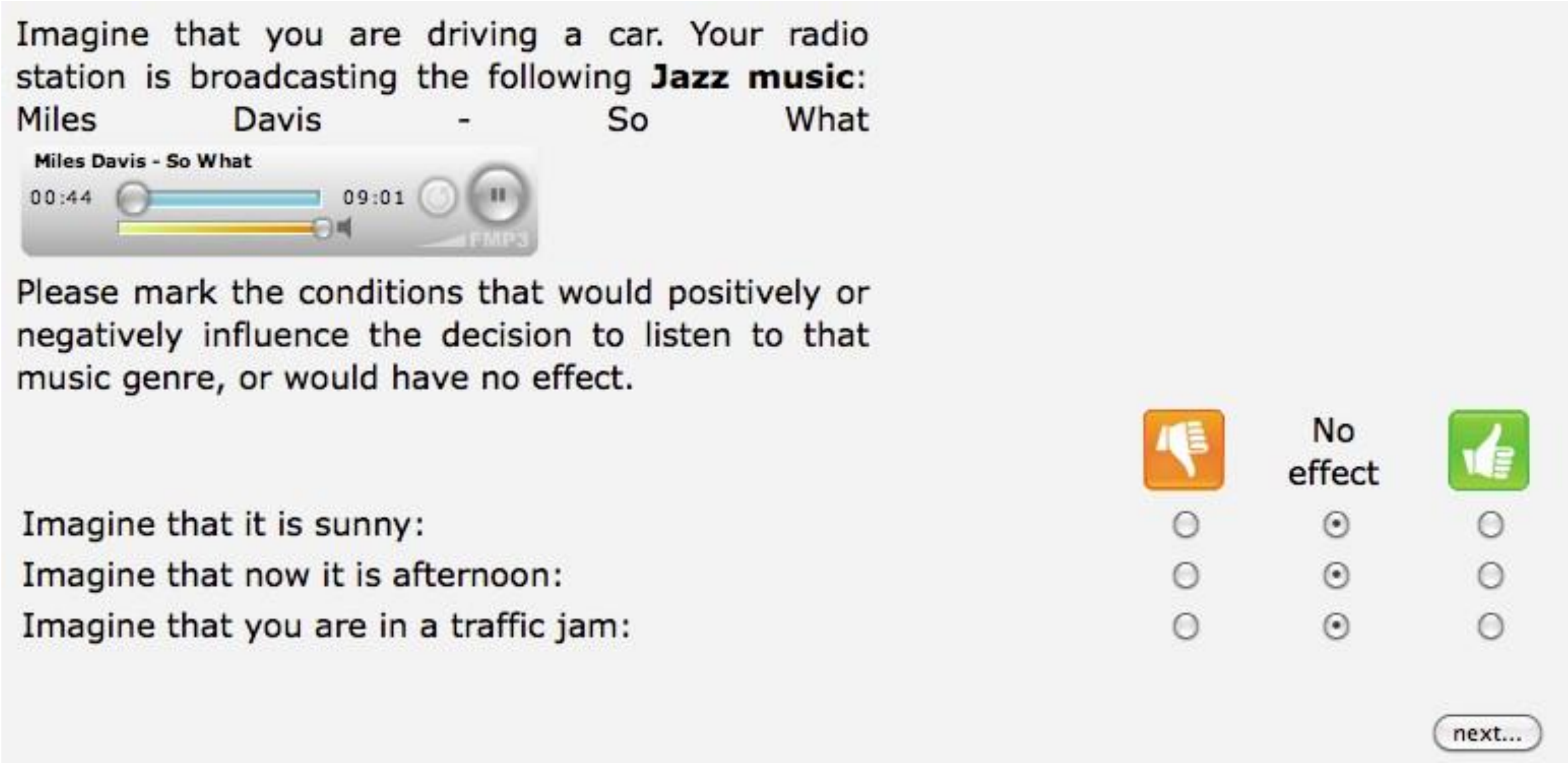
1. **Identifying potentially relevant contextual factors**
   - *Heuristics, consumer behavior literature*
2. **Ranking contextual factors**
   - *Based on subjective evaluations (what if scenario)*
3. **Measuring the dependency of the ratings from the contextual conditions and the users**
   - *Users rate items in imagined contexts*
4. **Modeling the rating dependency from context**
   - *Extended matrix factorization model*
5. **Learning the prediction model**
   - *Stochastic gradient descent*
6. **Delivering context-aware rating predictions and item recommendation**

[Baltrunas et al., 2012]

Contextual Factors

- **driving style (DS):** relaxed driving, sport driving
- **road type(RT):** city, highway, serpentine
- **landscape (L):** coast line, country side, mountains/ hills, urban
- **sleepiness (S):** awake, sleepy
- **traffic conditions (TC):** free road, many cars, traffic jam
- **mood (M):** active, happy, lazy, sad
- **weather (W):** cloudy, snowing, sunny, rainy
- **natural phenomena (NP):** day time, morning, night, afternoon

Determine Context Relevance



- Web based application
- We collected 2436 evaluations from 59 users

Expected Utility Estimation

User Study Results (I)

Blues	MI	Classical	MI	Country	MI	Disco	MI	Hip Hop	MI
driving style	0.32	driving style	0.77	sleepiness	0.47	mood	0.18	traffic conditions	0.19
road type	0.22	sleepiness	0.21	driving style	0.36	weather	0.17	mood	0.15
sleepiness	0.14	weather	0.09	weather	0.19	sleepiness	0.15	sleepiness	0.11
traffic conditions	0.12	natural phenomena	0.09	mood	0.13	traffic conditions	0.13	natural phenomena	0.11
natural phenomena	0.11	mood	0.09	landscape	0.11	driving style	0.10	weather	0.07
landscape	0.11	landscape	0.06	road type	0.11	road type	0.06	landscape	0.05
weather	0.09	road type	0.02	traffic conditions	0.10	natural phenomena	0.05	driving style	0.05
mood	0.06	traffic conditions	0.02	natural phenomena	0.04	landscape	0.05	road type	0.01

- Normalized Mutual Information of the contextual condition on the Influence variable (1/0/-1)
- The higher the MI the larger the influence

User Study Results (II)

Jazz	MI	Metal	MI	Pop	MI	Reggae	MI	Rock	MI
sleepiness	0.17	driving style	0.46	sleepiness	0.42	sleepiness	0.55	traffic conditions	0.24
road type	0.13	weather	0.26	driving style	0.34	driving style	0.38	sleepiness	0.22
weather	0.11	sleepiness	0.20	road type	0.27	traffic conditions	0.32	driving style	0.13
driving style	0.10	landscape	0.12	traffic conditions	0.23	mood	0.17	landscape	0.11
natural phenomena	0.08	traffic conditions	0.10	mood	0.14	landscape	0.15	road type	0.10
landscape	0.05	mood	0.07	natural phenomena	0.10	weather	0.13	mood	0.09
traffic conditions	0.05	road type	0.06	weather	0.07	natural phenomena	0.11	weather	0.08
mood	0.04	natural phenomena	0.05	landscape	0.05	road type	0.07	natural phenomena	0.08

- Normalized Mutual Information of the contextual condition on the Influence variable (1/0/-1)
- The higher the MI the larger the influence

Predictive Model

$$\hat{r}_{uic_1 \ldots c_k} = \boldsymbol{v}_u \cdot \boldsymbol{q}_i + \bar{\imath} + b_u + \sum_{j=1}^{k} b_{g_i j c_j}$$

- $\boldsymbol{v}_u$ and $\boldsymbol{q}_i$ are d dimensional real valued vectors representing the user *u* and the item *i*
- $\bar{\imath}$ is the average of the item *i* ratings
- b_u is a baseline parameter for user u
- b_{gjc} is the baseline of the **contextual condition** c_j (factor *j*) and **genre** g_i of item *i*
  - *We assume that context influences uniformly all the tracks with a given genre*
- If a contextual factor is unknown, i.e., $c_j = 0$, then the corresponding baseline b_{gjc} is set to 0.

Training the Model

$$\min_{v_*,q_*,b_*} \sum_{r\in R} [(r_{uic_1\ldots c_k} - \boldsymbol{v}_u \cdot \boldsymbol{q}_i - \bar{\imath} - \sum_{j=1}^{k} b_{g_i j c_j})^2 + \lambda(\|\boldsymbol{v}_u\|^2 + \|\boldsymbol{q}_i\|^2 + \sum_{j=1}^{k} b^2_{g_i j c_j})]$$

- Added regularization to avoid over fitting
- We use the stochastic gradient descent method for fast training
- Linear time complexity in the amount of data and in the number of contextual conditions

Modeling Context-Item dependencies

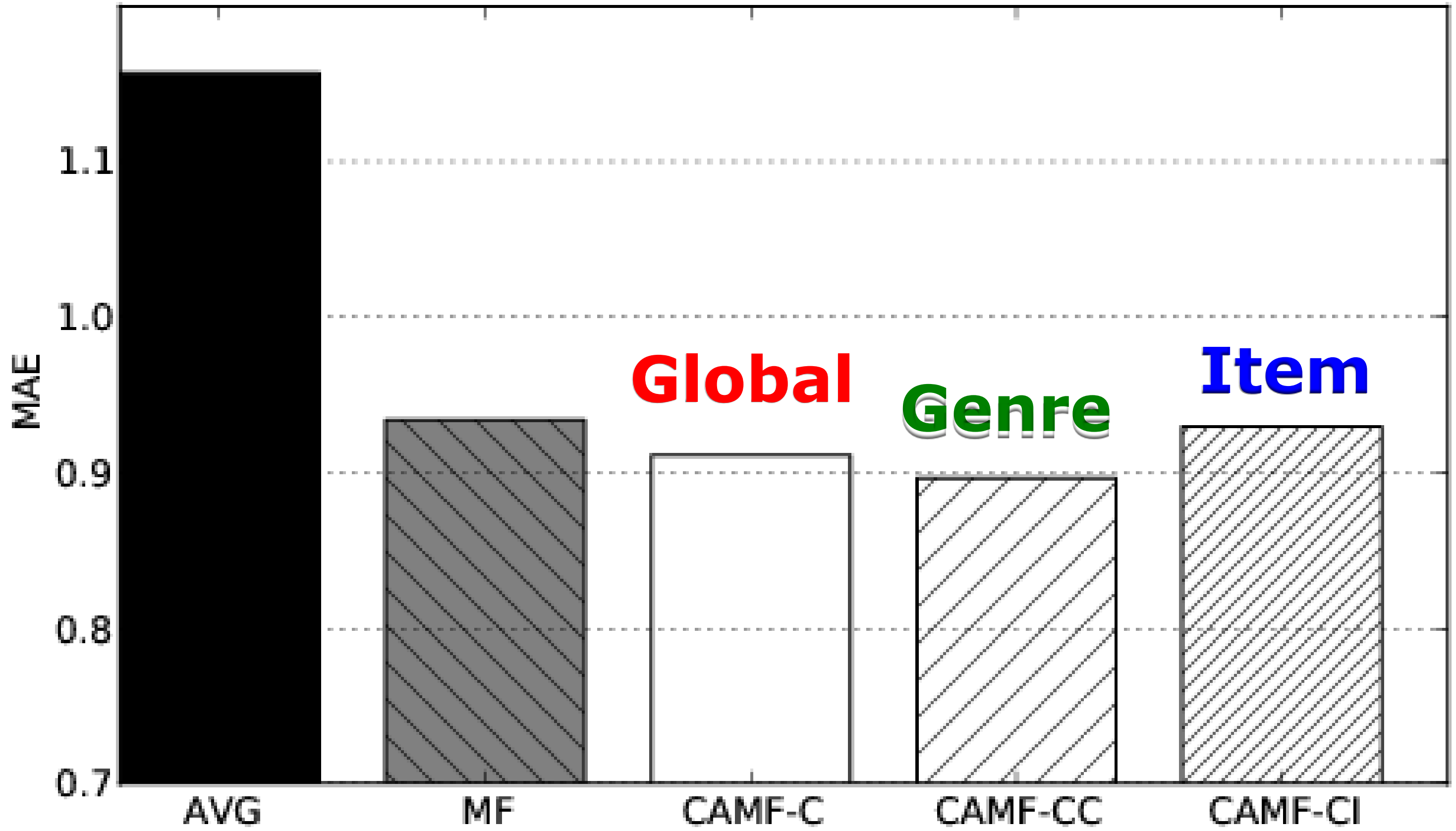
- CAMF-C assumes that each contextual condition has a **global** influence on the ratings - independently from the item
- CAMF-CC introduces one model parameter for each contextual condition and item **category** (music genre) – as shown before
- CAMF-CI introduces one parameter per each contextual condition and **item** pair

Global

Genre

Item

Predicting Expected Utility in Context



Item average

Matrix Factorization (personalization)

Matrix Factorization and context

[Baltrunas et al., 2011]

Major obstacle for contextual computing

- **Understand** the impact of contextual dimensions on the personalization process
- **Selecting** the right information, i.e., relevant in a particular personalization task
- **Obtain sufficient** and **reliable data** describing the user preferences in context
- **Embed** the contextual dimension in a more classical – simpler - recommendation **computational model**.

Summary

- There is no rating without context – context let us understand the circumstances
- Context modeling requires a multidimensional rating function
  - Sparcity of the available samples
  - Simple data mining approaches cannot work properly
  - Several prediction tasks are possible
  - There is space for multiple prediction methods
- Context changes during the interaction with the recommender system – it should be taken into account to adapt the next stages.

References

- G. Adomavicius, R. Sankaranarayanan, S. Sen, A. Tuzhilin. Incorporating contextual information in recommender systems using a multidimensional approach. ACM TOIS, 23(1):103–145, 2005.
- G. Adomavicius, B. Mobasher, F. Ricci, and A. Tuzhilin. Context-aware recommender systems. AI Magazine, 32(3):67–80, 2011.
- L. Baltrunas, F. Ricci. Context-based splitting of item ratings in collaborative filtering. RecSys 2009: 245-248, 2009.
- L. Baltrunas, M. Kaminskas, B. Ludwig, O. Moling, F. Ricci, A. Aydin, K.-H. Lueke, and R. Schwaiger. Incarmusic: Context-aware music recommendations in a car. In E-Commerce and Web Technologies - 12th International Conference, EC-Web 2011, Toulouse, France, August 30 - September 1, 2011. Proceedings, pages 89–100. Springer, 2011.
- L. Baltrunas, B. Ludwig, S. Peer, and F. Ricci. Context relevance assessment and exploitation in mobile recommender systems. Personal and Ubiquitous Computing, pages 1–20, 2011.
- T. Mahmood, F. Ricci. Improving recommender systems with adaptive conversational strategies. Hypertext09: 73-82, 2009.
- J. E. Pitkow, H. Schütze, T. A. Cass, R. Cooley, D. Turnbull, A. Edmonds, E. Adar, T. M. Breuel: Personalized search. Commun. ACM 45(9): 50-55, 2002.