PES UNIVERSITY ONLINE

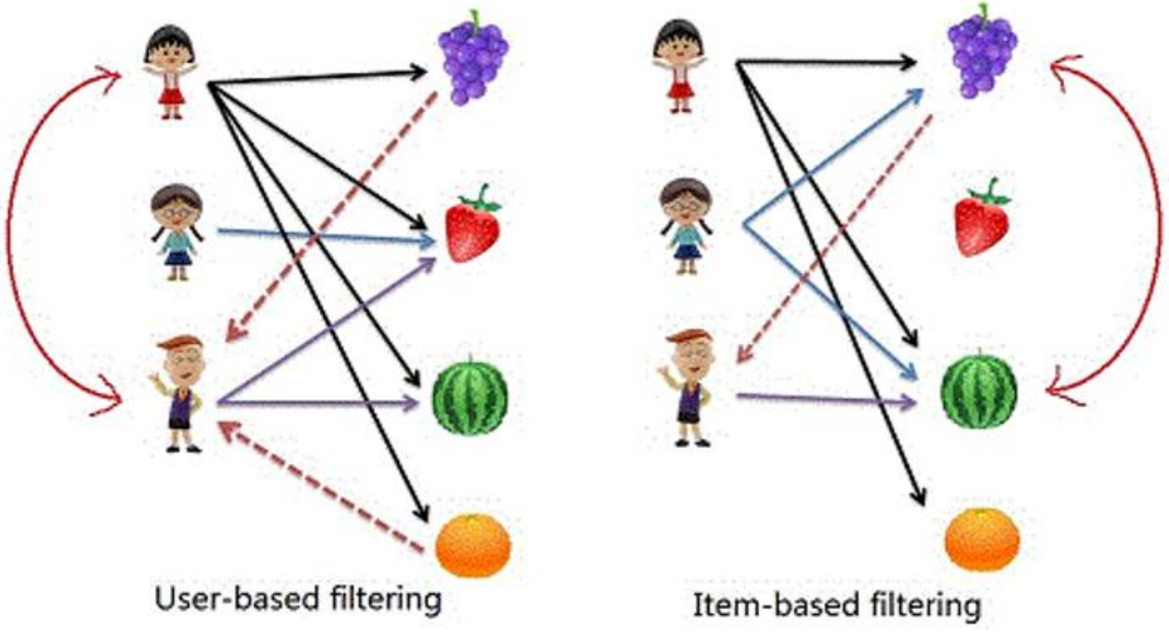
DATA ANALYTICS

Unit 4: Recommendation Systems

Jyothi R.
Department of Computer Science and Engineering

Amazon's Item-to-Item CF

Difference with User-to-User CF



Amazon’s Item-to-Item CF

Similarity of user i with item 17

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
,1	,3	,6	,1	,3	,4	,3	,3	,2	,6	,2	,5	,4	,5	,5	,3	1	,3	,5	,4	,2	,4	,4	,5

Items

Users

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
a			1		4	5			4		3					2			4		2				
b			4							3							5	1		3					
c		5		4			4						3		5					4		5			
d								3				5				3			4		2			3	
e		3					5			4	5				5					1			5	4	
f			4				1		3	5		4	1		5	4	4		4				3		
g	2	4				4		2			5		1	4	5		4	2	4		5			4	
h			2		1		4		3	5		4	2		5	4	5						5		
I		1					3			5				5		4	4		5			4		3	
j			4			4				5			1		5		4		4				4		
k		5				4			2		5		1	5		4		2		4				2	
l					3			3				4	1		4		4	2	4					3	
m	5		3					5	3		5	4		5	5	3			4	4	5	4		4	
n			1		4	5				4	5		1	5		4		3		4		4	3		
o			4			4				5		4		5			4	2		5		5		3	
p				4			5								5	4		2	4	4	5	4		2	
q					3			3					1	5		4	4		4			4		3	
r		4			1	4		2					2		5		4				5	4		4	
s			2		4		4			5			1			4		2	4		4		5		
t		1		4			3					4		5	5		4			4				3	
u			2		1		4		3				1		5	4		2	4		5	4			
v					4	5				4	3		5			2					2			5	
w				2			2		3			5			4	5		4	2		3	4			
x	4			5				3		3				4	5					1					
y			1			3				2	3						3	3		5		4			

Amazon' s Item-to-Item CF

- How it works
- Matches each of the user's purchased and rated items to similar items
- Combines those similar items into a recommendation list

An iterative algorithm:

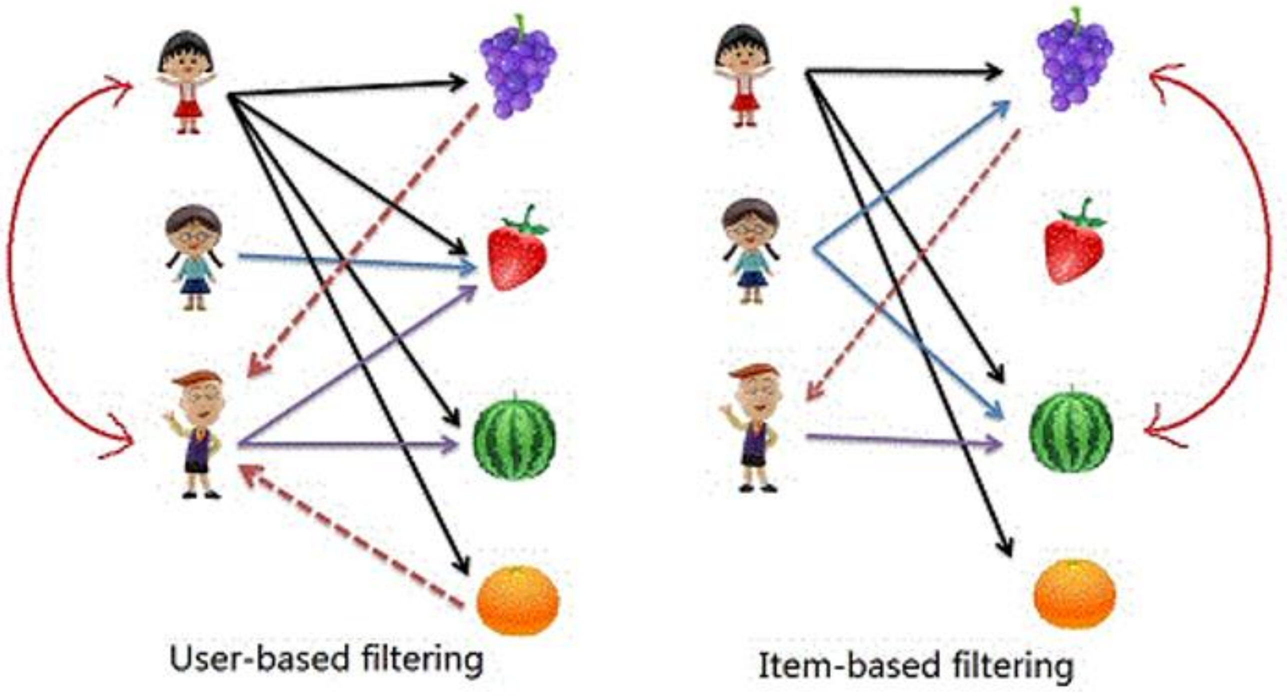
- Builds a similar-items table by finding items that customers tend to purchase together
- Provides a better approach by calculating the similarity between a single product and all related products:

Amazon's Item-to-Item CF

- The similarity between two items uses the cosine measure
- Each Mx1 vector corresponds to an item
- A vector's M dimensions correspond to customers who have purchased that item

```
For each item in product catalog, I1
  For each customer C who purchased I1
    For each item I2 purchased by customer C
      Record that a customer purchased I1 and I2
  For each item I2
    Compute the similarity between I1 and I2
```

Amazon's Item-to-Item CF vs User-based CF



Item-Item vs. User-User Scalability and Quality: Comparison

Item-to-Item collaborative filtering:

- scalability and performance are achieved by creating the expensive similar-items table offline
- online component "looking up similar items“ scales independently of the catalog size or the number of customers
- fast for extremely large data sets
- recommendation quality is excellent since it recommends highly correlated similar items
- unlike traditional collaborative filtering,
  - the algorithm performs well with limited user data,
  - producing high-quality recommendations based on as few as two or three items.

User Based collaborative filtering:

- little or no offline computation
- impractical on large data sets, unless it uses dimensionality reduction, sampling, or partitioning
- dimensionality reduction, sampling, or partitioning reduces recommendation quality

Cluster models:

- can perform much of the computation offline
- but recommendation quality is relatively poor

Results:

- The MovieLens dataset contains 1 million ratings from 6,040 users on 3,900 movies.
- The best overall results are reached by the item-by-item based approach. It needs 170 seconds to construct the model and 3 seconds to predict 100,021 ratings.

	User Based	Model Based	Item Based
Model Construction Time (sec.)	730	254	170
Prediction Time (sec.)	31	1	3
MAE	0.6688	0.6736	0.6382

“Core” recommender systems

The conceptual goals of various recommender systems

Approach	Conceptual Goal	Input
Collaborative	Gives us recommendations based on a collaborative approach that leverages the ratings and actions of our peers/myself	User ratings + Community ratings
Content-based	Gives us recommendations based on the content (attributes) we have favored in our past ratings and actions.	User ratings + item attributes + domain knowledge
Knowledge-based	Gives us recommendations based on our explicit specification of the kind of content (attributes) we want	User specification + Item attributes + domain knowledge

"Core" Recommendation Techniques

Technique	Background	Input	Process
Collaborative	Ratings from U of items in I	Ratings from U of items in I	Identify users in U similar to u, and extrapolate from their ratings of i
Content-based	Features of items in I	U's ratings of items in I	Generate a classifier that fits U's rating behavior and use it on I
Demographic	Demographic information about U and their ratings of items in I	Demographic information about U	Identify users that are demographically similar to U, ad extrapolate from their ratings of i
Utility-based	Features of items in I	A Utility function over items in I that describes U's preferences.	Apply the function to the items and determine I's rank
Knowledge-based	Features of items in I. Knowledge of how these items meet a user's needs.	A description of U's needs or interests.	Infer a match between I and U's need.

Knowledge-based recommender systems

Knowledge-based recommender systems are appropriate in the following situations:

1. Customers want to explicitly specify their requirements. Therefore, interactivity is a crucial component of such systems. Note that collaborative and content-based systems do not allow this type of detailed feedback.

2. It is difficult to obtain ratings for a specific type of item because of the greater complexity of the product domain in terms of the types of items and options available.

3. In some domains, the ratings may be time-sensitive. The ratings on an old car or computer are not very useful for recommendations because they evolve with changing product availability and corresponding user requirements.

Knowledge-based recommender systems types

- Knowledge-based recommender systems can be categorized on the basis of user interactive methodology and the corresponding knowledge bases used to facilitate the interaction.

- There are two primary types of knowledge-based recommender systems:

1. Constraint-based recommender systems: In constraint-based systems users typically specify requirements or constraints (e.g., lower or upper limits) on the item attributes. Furthermore, domain-specific rules are used to match the user requirements or attributes to item attributes. These rules represent the domain-specific knowledge used by the system.

2. Case-based recommender systems: In case-based recommender systems, specific cases are specified by the user as targets or anchor points. Similarity metrics are defined on the item attributes to retrieve similar items to these targets. The similarity metrics are often carefully defined in a domain-specific way.

Knowledge-Based Recommender Systems

- The Interaction between user and recommender may take the following forms.

1. **Conversational Systems**: The user preferences are determined in the context of a feedback loop. The item domain is complex, and the user preferences can be determined only in the context of an iterative conversational system.

2. **Search-based systems**: User preferences are elicited by using a preset sequence of questions such as the following;” Do you prefer a house in a suburban area or within the city?”

3. **Navigation-based recommendation**: The user specifies a number of change requests to item being currently recommended. Through an iterative set of change requests, it is possible to arrive at a desirable item.

   Eg. “ I would like a similar house about 5 miles west of the currently recommended house.”

Such recommender systems are also referred to as critiquing recommender systems.

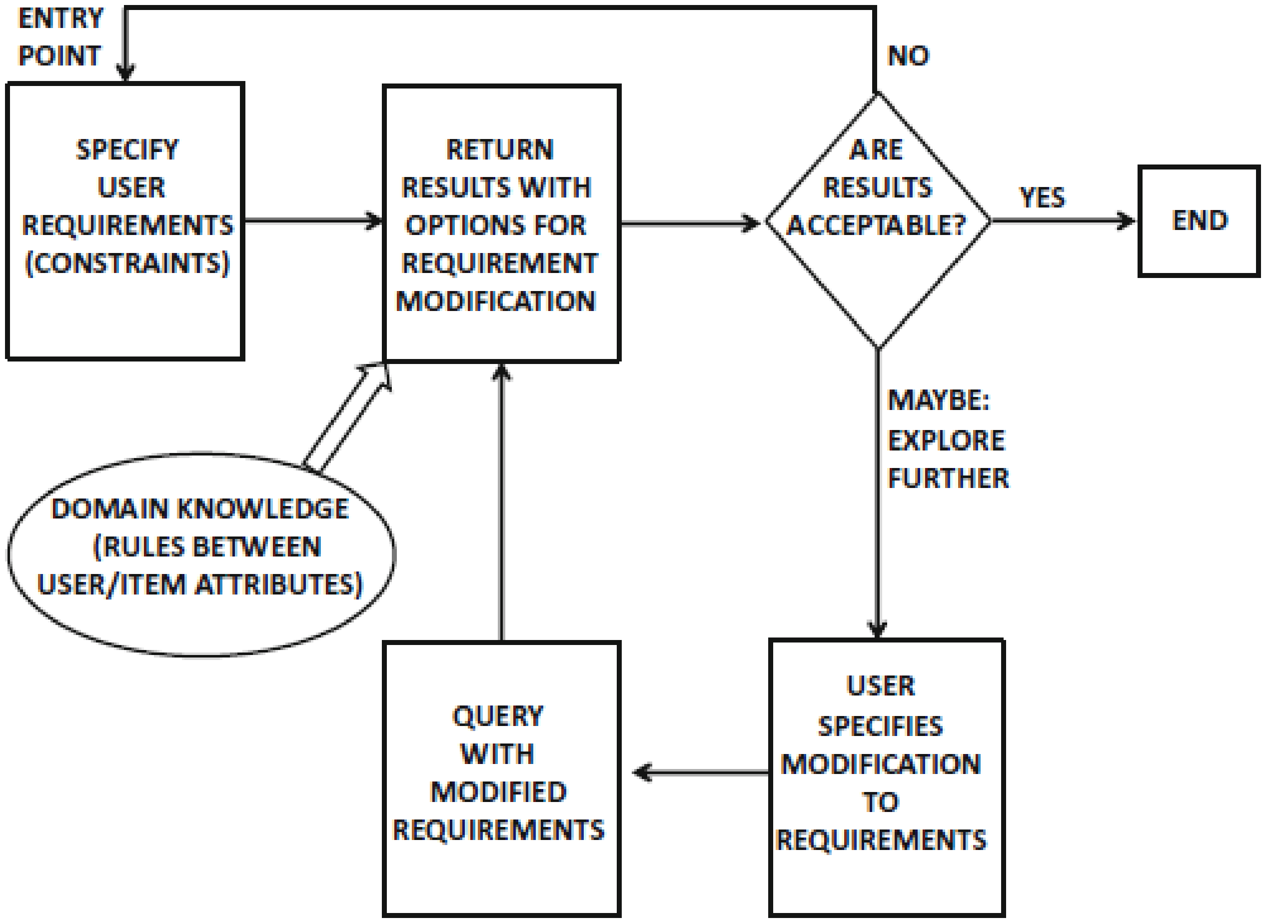
Knowledge-Based Recommender Systems

- Critiquing recommender systems are naturally designed for case-based recommender systems, because one critiques a specific case in order to arrive at the desired outcome.

- A search-based system can be used to set up user requirements for constraint-based recommenders.

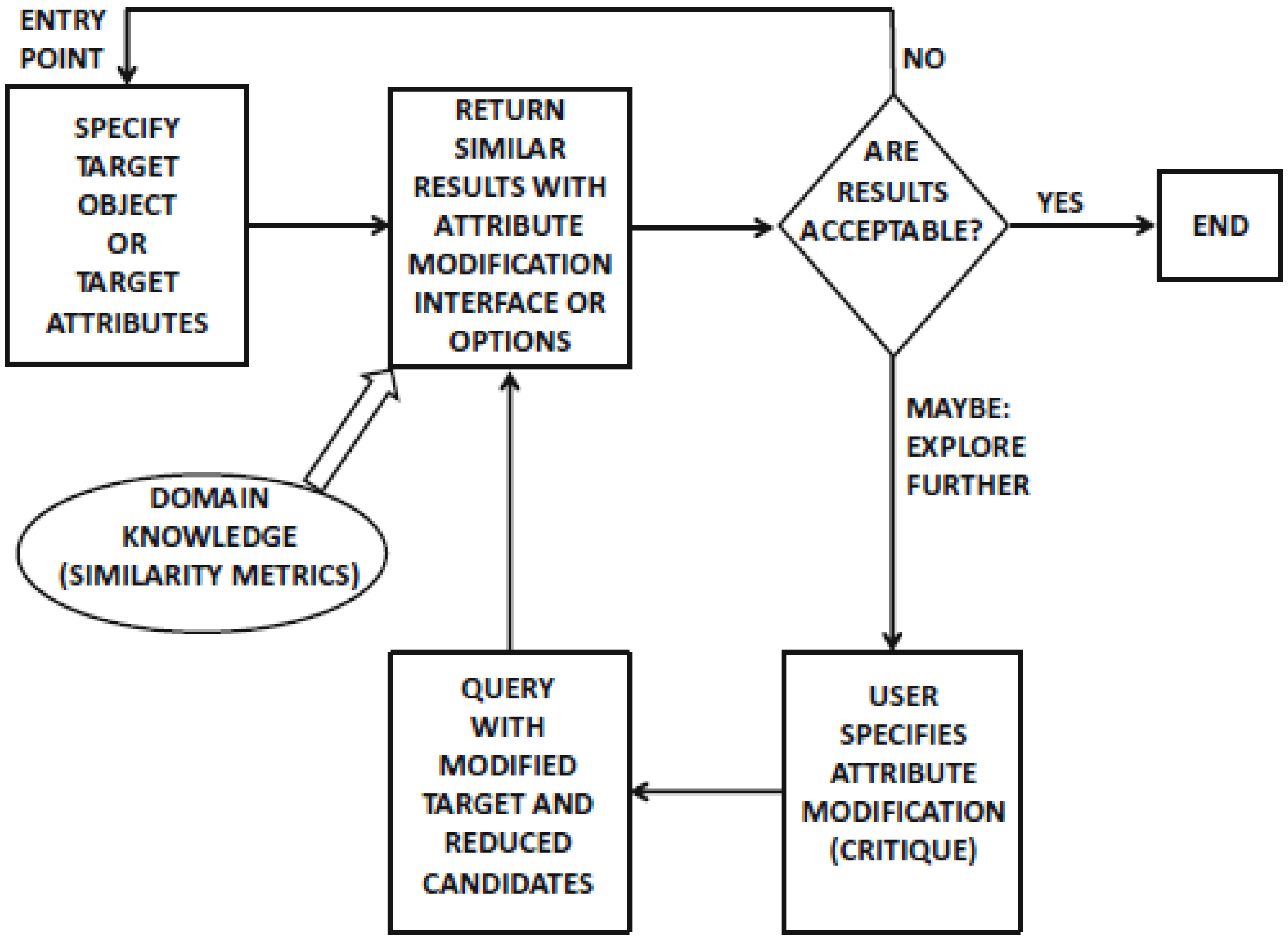
Knowledge-based Recommendation system

Constraint-based Recommender systems.



Knowledge-based Recommendation system

Case-based Recommender systems.



Knowledge-based Recommendation system

Difference between Constraint-based and Case-based Recommender systems.

- In constraint-based systems, specific requirements or constraints are specified by the user.
- The Original query is modified by addition, deletion, modification, or relaxation of the original set of user requirements.
- Users are not in a position to exactly state their requirements up front in a complex product domain, this problem is partially addressed through a knowledge-base of rules, which map user requirements to product attributes.

- In case-based systems, specific targets or cases are specified.
- Either the target is modified through user interaction, or the search results are pruned through the use of directional critiques.
- This problem is addressed through a conversational style of critiquing.

Knowledge-based recommender systems types

Examples of attributes in a recommendation application for buying homes.

Item-Id	Beds	Baths	Locality	Type	Floor Area	Price
1	3	2	BTM	Town House	1600	220,000
2	5	2.5	JP	Split-level	3600	973,000
3	4	2	RT	Ranch	2600	630,000
4	2	1.5	MAJESTIC	Condo	1500	220,000
5	4	2	Dollars	Colonial	2700	430,000

Knowledge-Based Recommender Systems

- Suggests products based on **inferences** about a user`s needs and preferences

- **Functional knowledge:** about how a particular item meets a particular user need

- The **user model** can be any knowledge structure that supports this inference

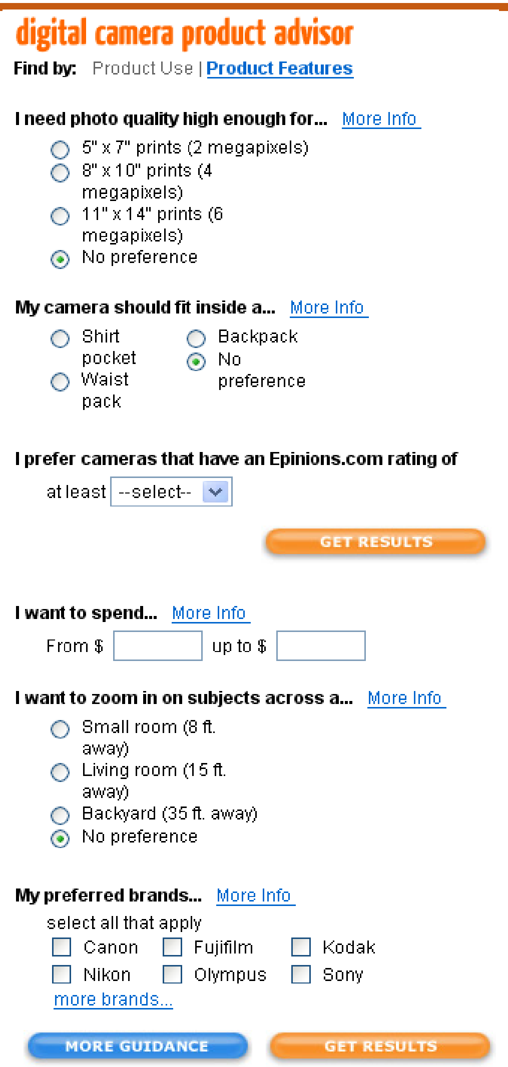
- A query, i.e., the set of preferred features for a product
- A case (in a case-based reasoning system)
- An adapted similarity metric (for matching)
- A part of an ontology

- **There is a large use of domain knowledge encoded in a knowledge representation language/approach.**

DATA ANALYTICS

Knowledge-Based Recommender Systems

PES UNIVERSITY ONLINE



digital camera product advisor

Find by: Product Use | Product Features

I need photo quality high enough for... More Info
- 5" x 7" prints (2 megapixels)
- 8" x 10" prints (4 megapixels)
- 11" x 14" prints (6 megapixels)
- No preference

My camera should fit inside a... More Info
- Shirt pocket
- Waist pack
- Backpack
- No preference

I prefer cameras that have an Epinions.com rating of

at least --select--

GET RESULTS

I want to spend... More Info

From $ up to $

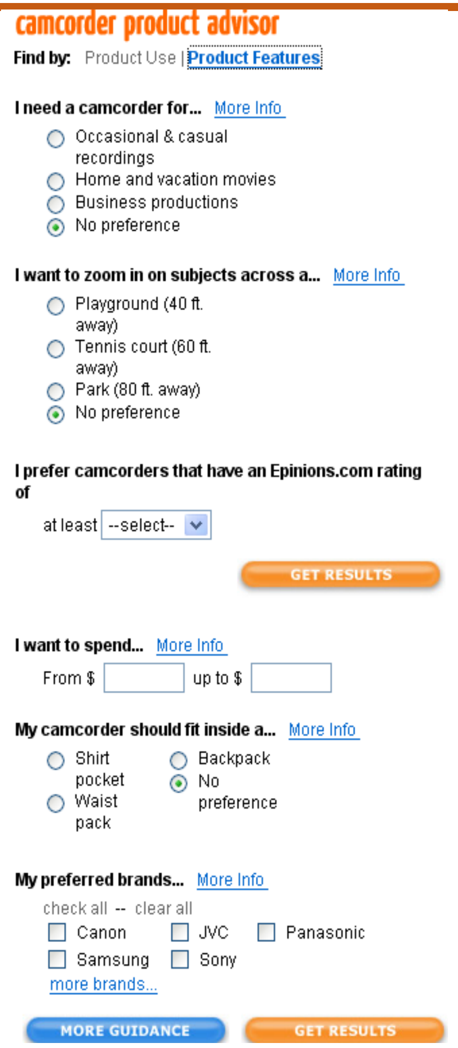
I want to zoom in on subjects across a... More Info
- Small room (8 ft. away)
- Living room (15 ft. away)
- Backyard (35 ft. away)
- No preference

My preferred brands... More Info

select all that apply
- Canon
- Fujifilm
- Kodak
- Nikon
- Olympus
- Sony

more brands...

MORE GUIDANCE GET RESULTS



camcorder product advisor

Find by: Product Use | Product Features

I need a camcorder for... More Info
- Occasional & casual recordings
- Home and vacation movies
- Business productions
- No preference

I want to zoom in on subjects across a... More Info
- Playground (40 ft. away)
- Tennis court (60 ft. away)
- Park (80 ft. away)
- No preference

I prefer camcorders that have an Epinions.com rating of

at least --select--

GET RESULTS

I want to spend... More Info

From $ up to $

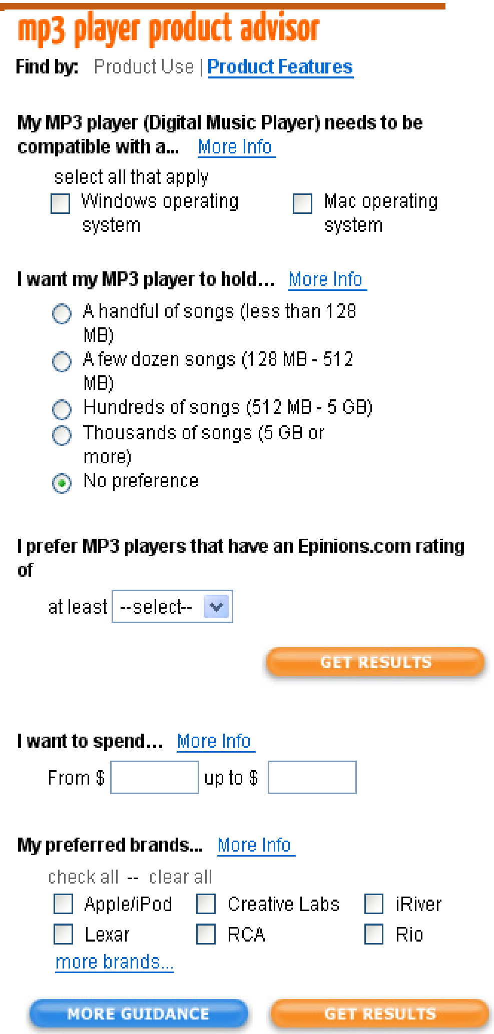
My camcorder should fit inside a... More Info
- Shirt pocket
- Waist pack
- Backpack
- No preference

My preferred brands... More Info

check all -- clear all
- Canon
- JVC
- Panasonic
- Samsung
- Sony

more brands...

MORE GUIDANCE GET RESULTS



mp3 player product advisor

Find by: Product Use | Product Features

My MP3 player (Digital Music Player) needs to be compatible with a... More Info

select all that apply
- Windows operating system
- Mac operating system

I want my MP3 player to hold... More Info
- A handful of songs (less than 128 MB)
- A few dozen songs (128 MB - 512 MB)
- Hundreds of songs (512 MB - 5 GB)
- Thousands of songs (5 GB or more)
- No preference

I prefer MP3 players that have an Epinions.com rating of

at least --select--

GET RESULTS

I want to spend... More Info

From $ up to $

My preferred brands... More Info

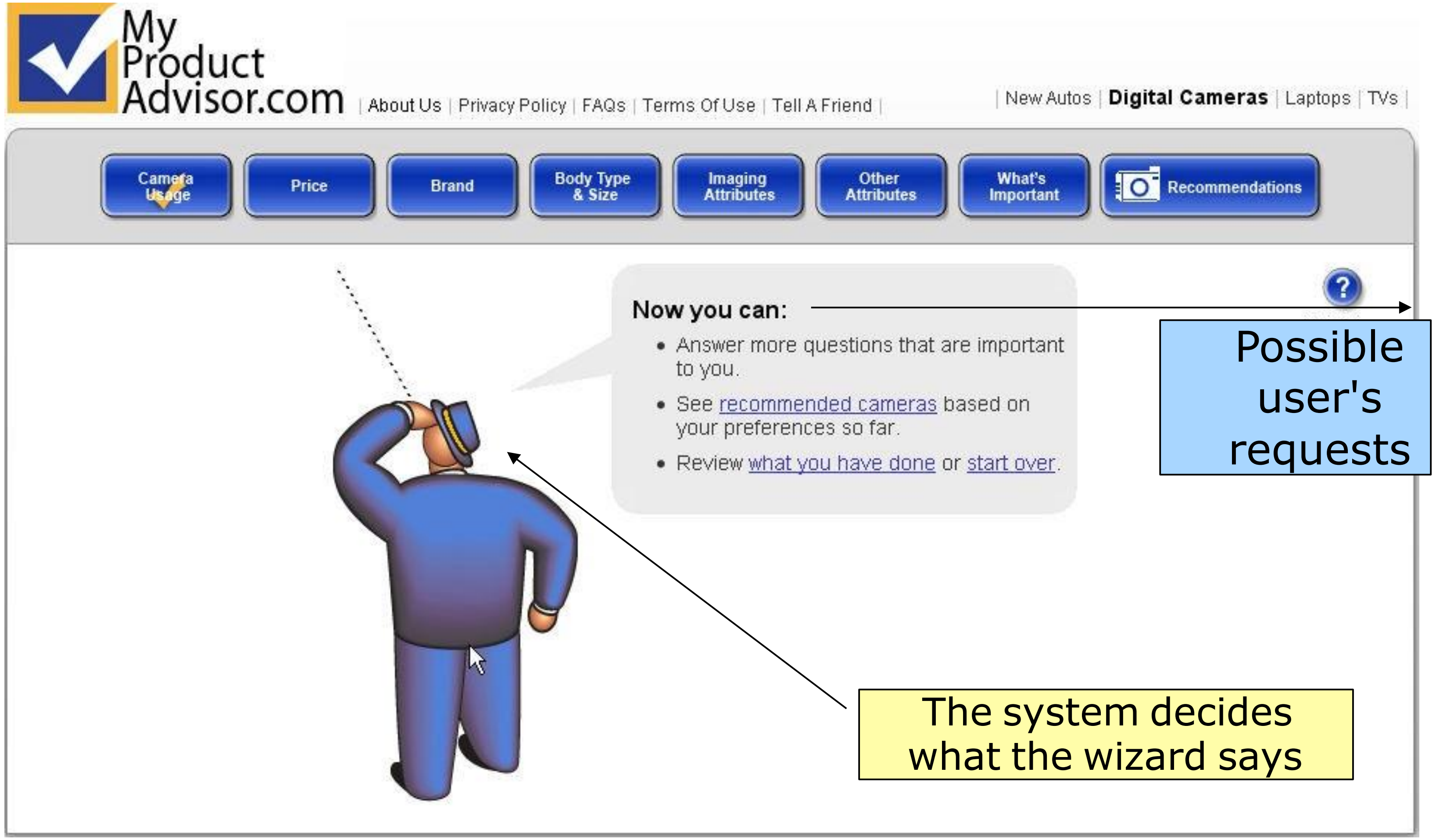
check all -- clear all
- Apple/iPod
- Creative Labs
- iRiver
- Lexar
- RCA
- Rio

more brands...

MORE GUIDANCE GET RESULTS

ActiveBuyersGuide

Knowledge-Based Recommender Systems

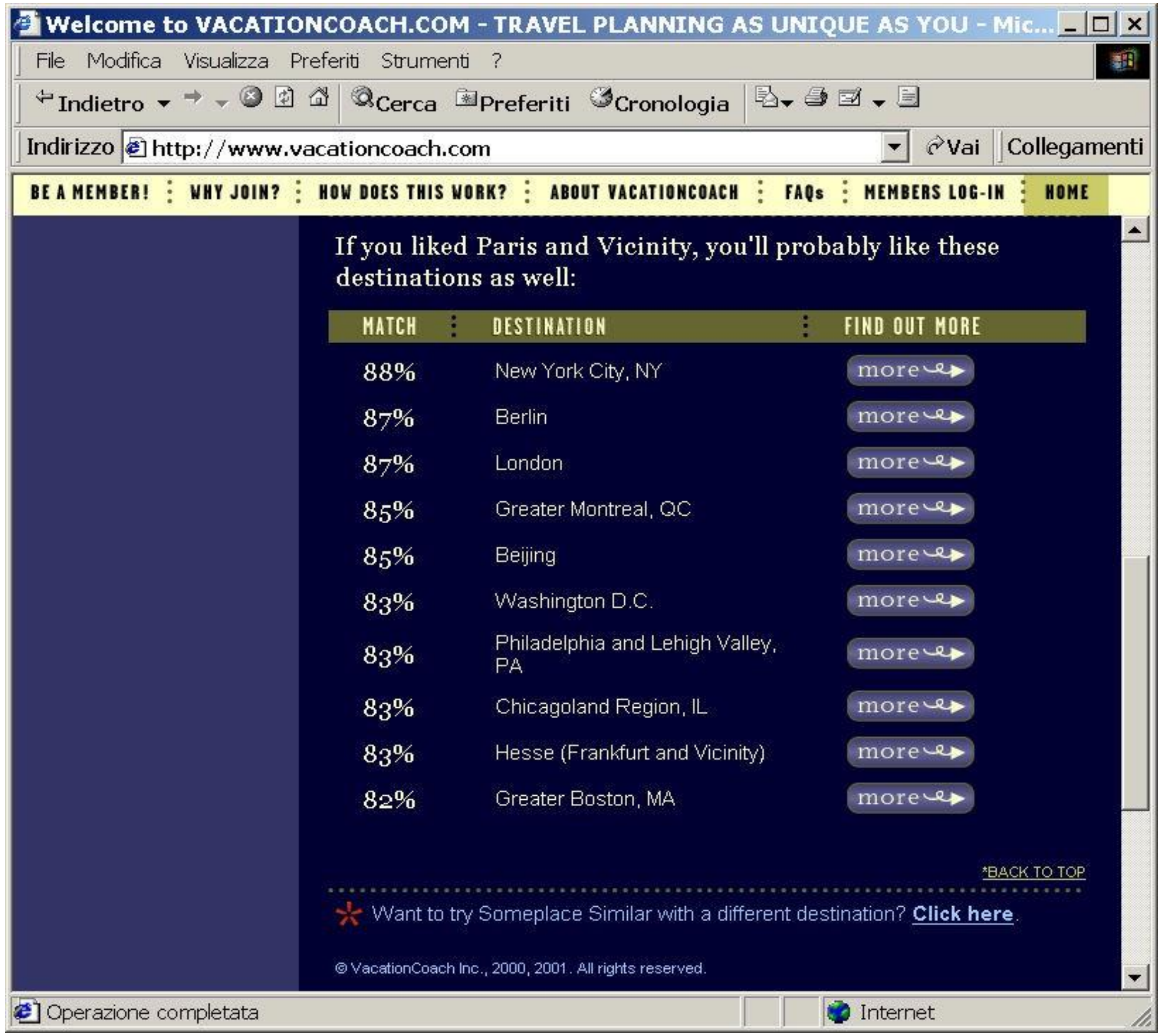


Knowledge-Based Recommender Systems

Knowledge-Based Recommender Systems

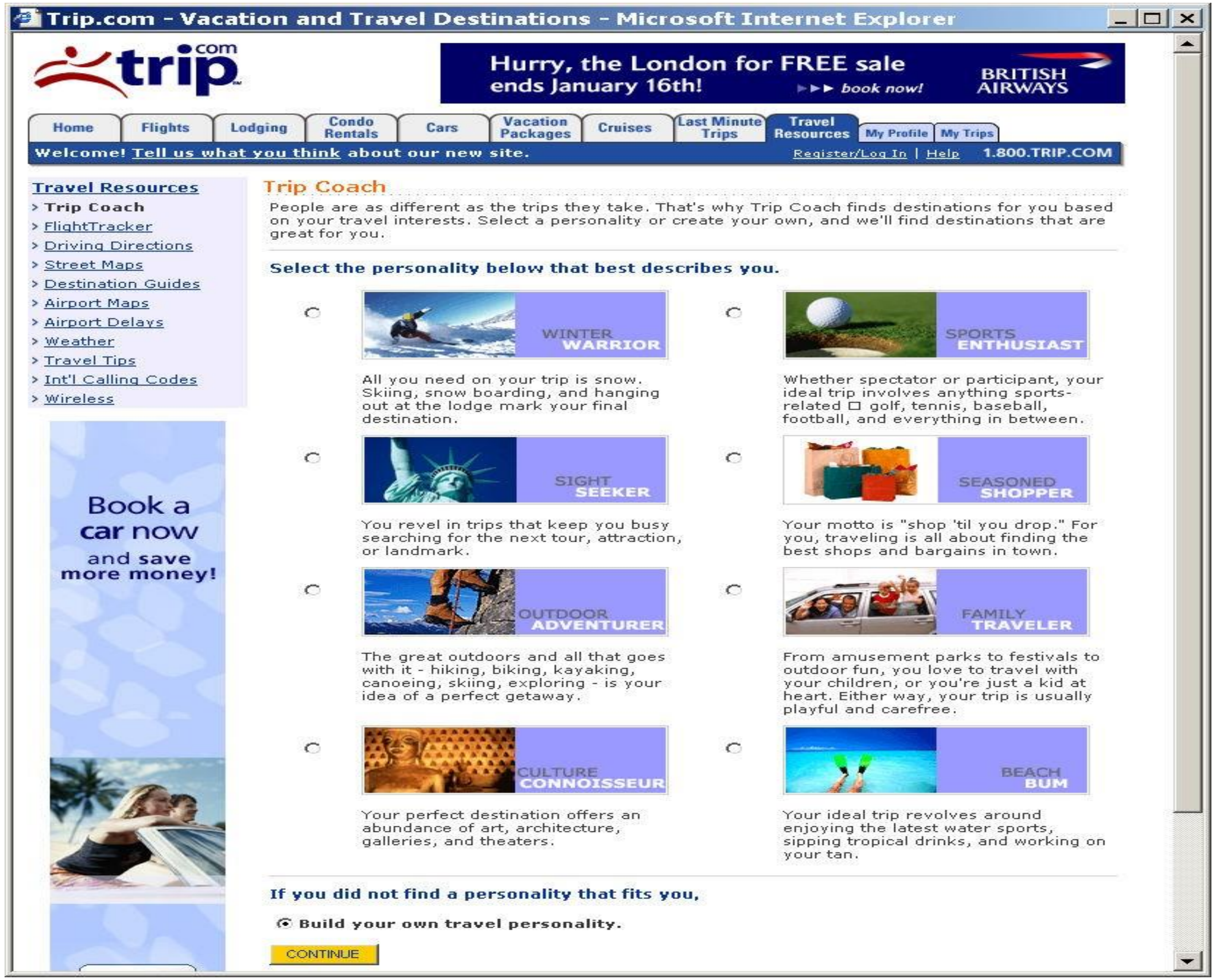
DATA ANALYTICS

Knowledge-Based Recommender Systems



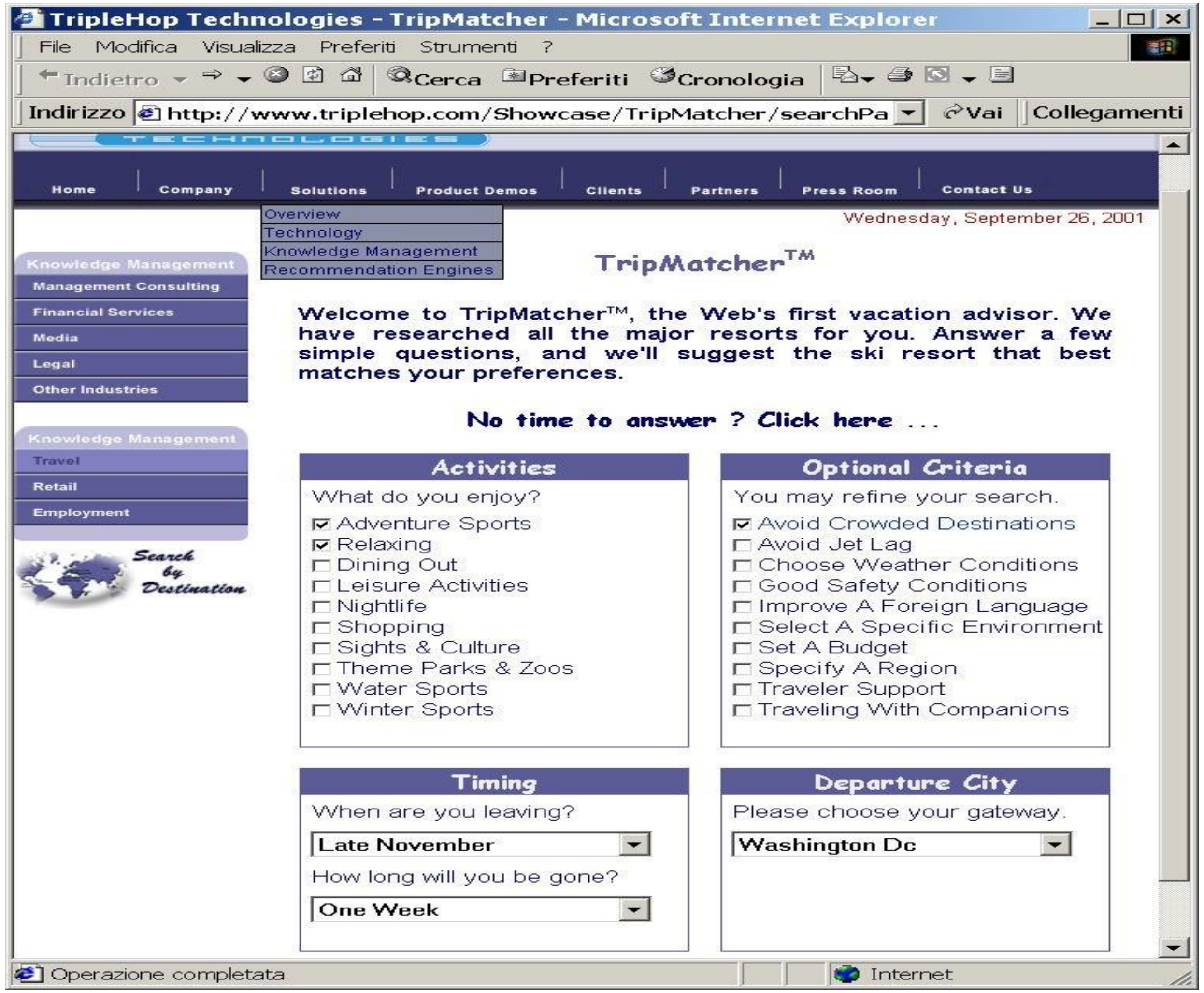
Knowledge-Based Recommender Systems

Knowledge-Based Recommender Systems



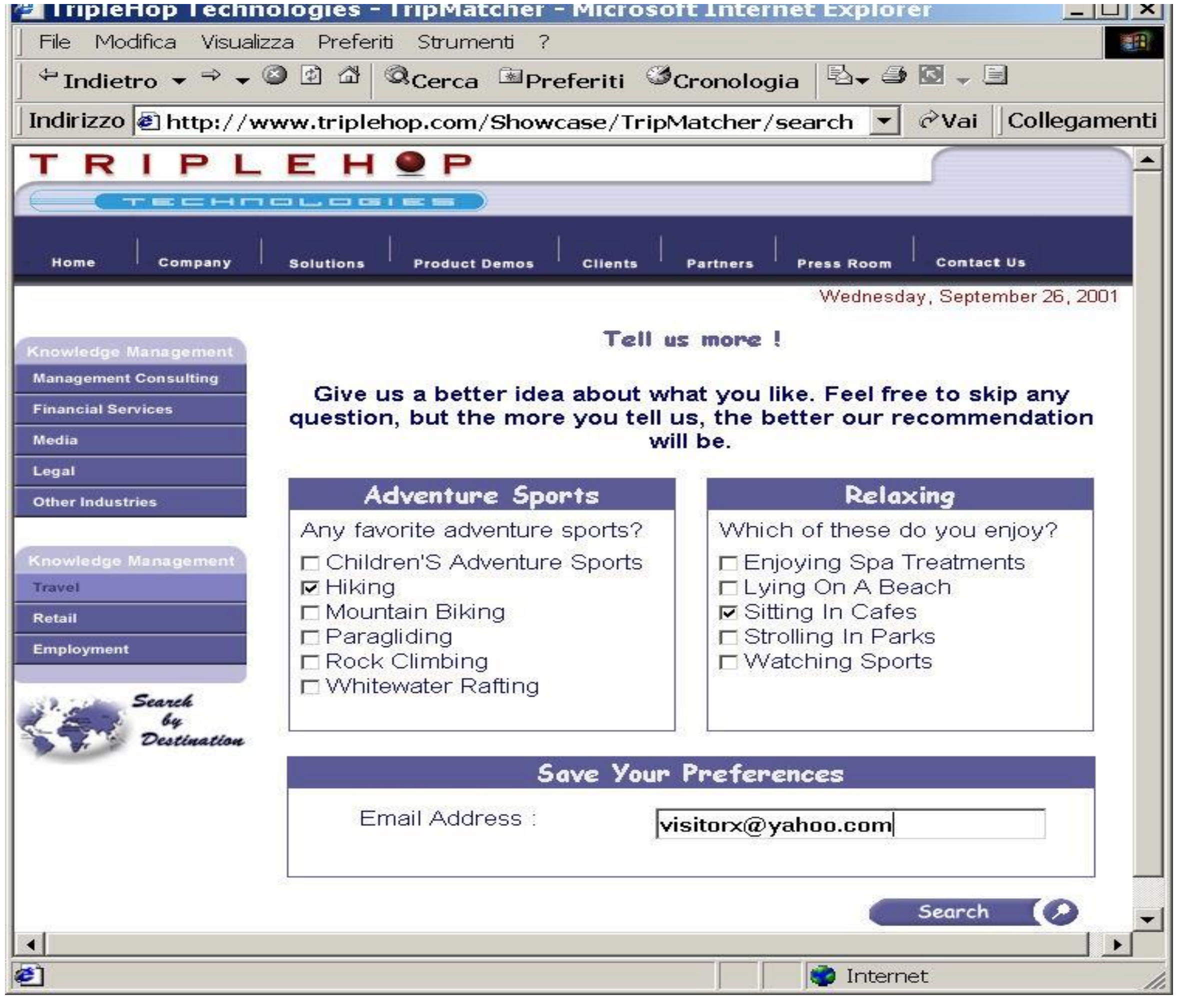
Trip.com

Knowledge-Based Recommender Systems



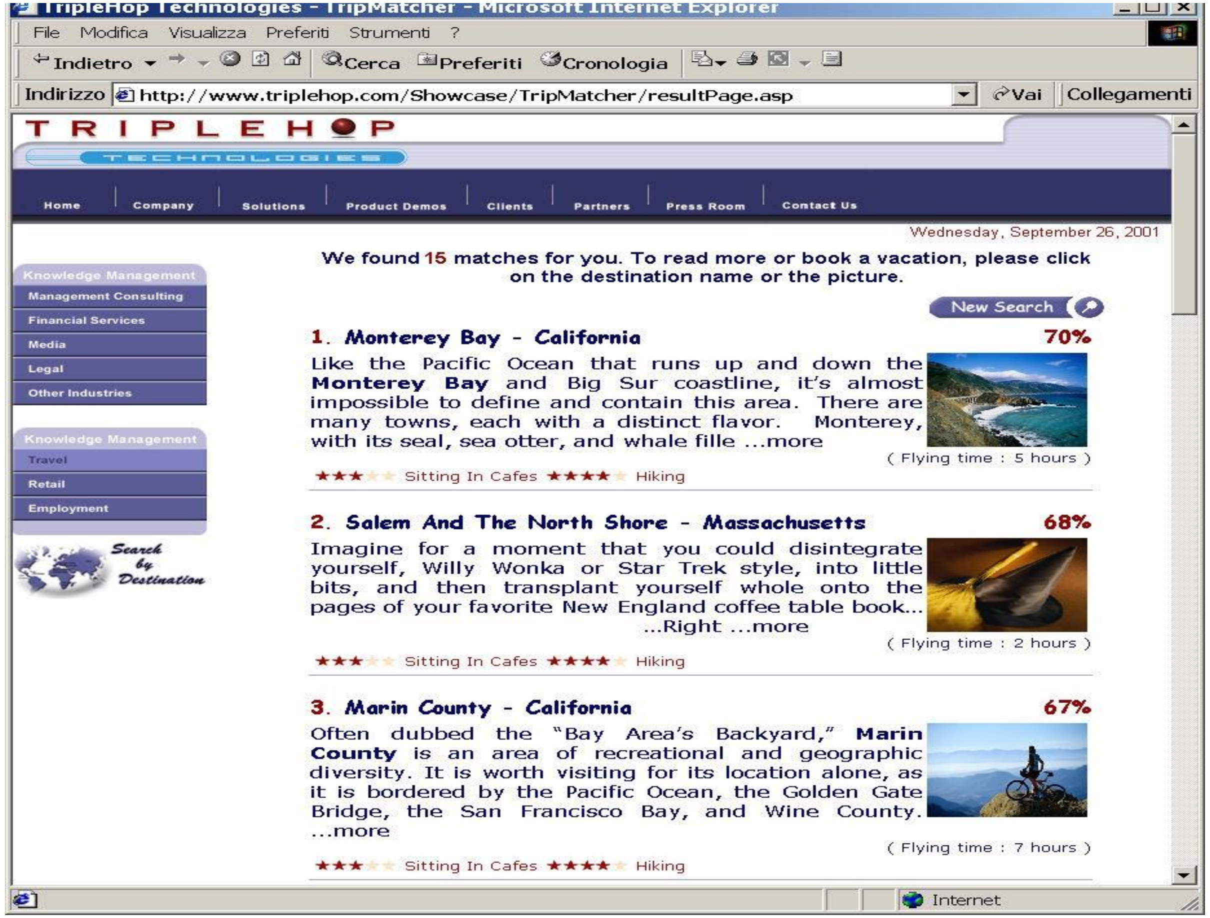
Trip.com

Knowledge-Based Recommender Systems



Trip.com

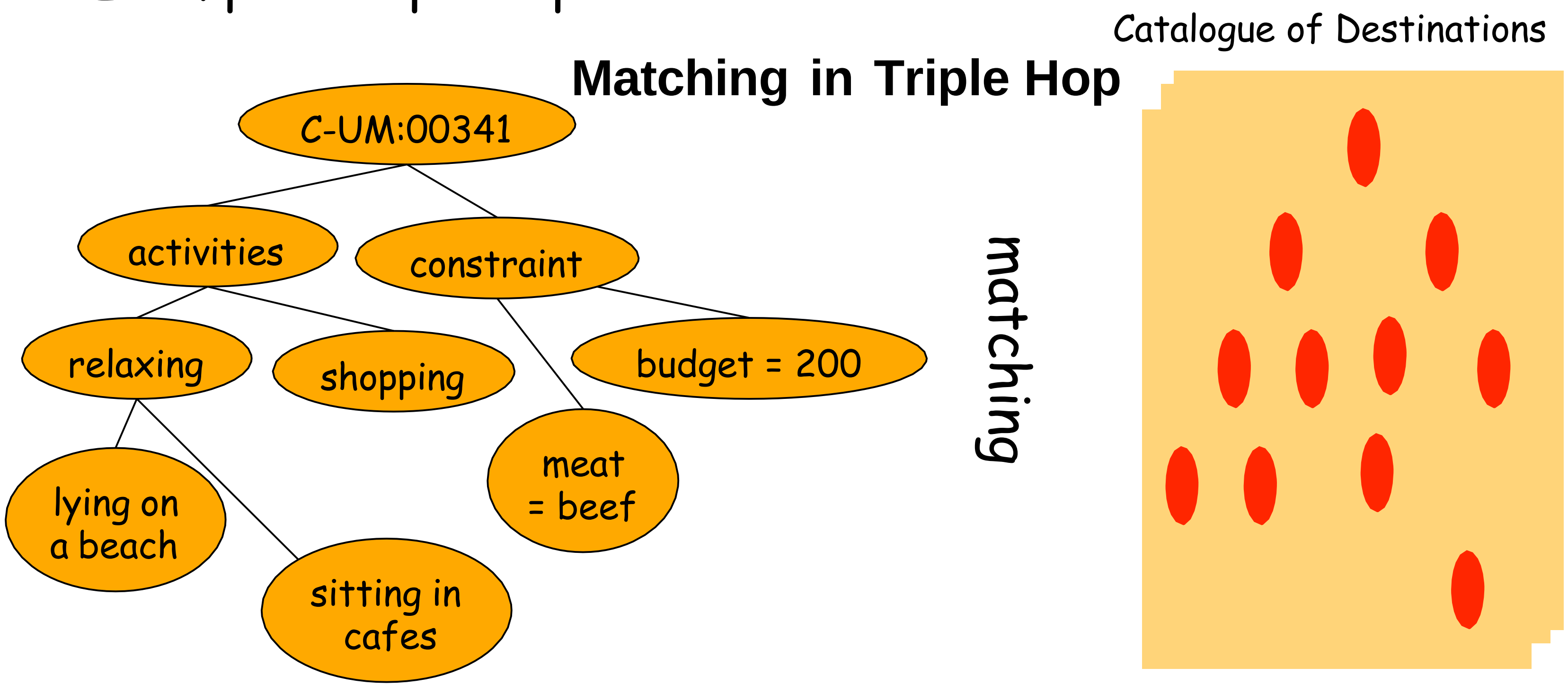
Knowledge-Based Recommender Systems



Trip.com

Knowledge-Based Recommender Systems

Example: TripleHop



Knowledge-Based Recommender Systems

PES UNIVERSITY ONLINE

TripleHop and Content-Based RS

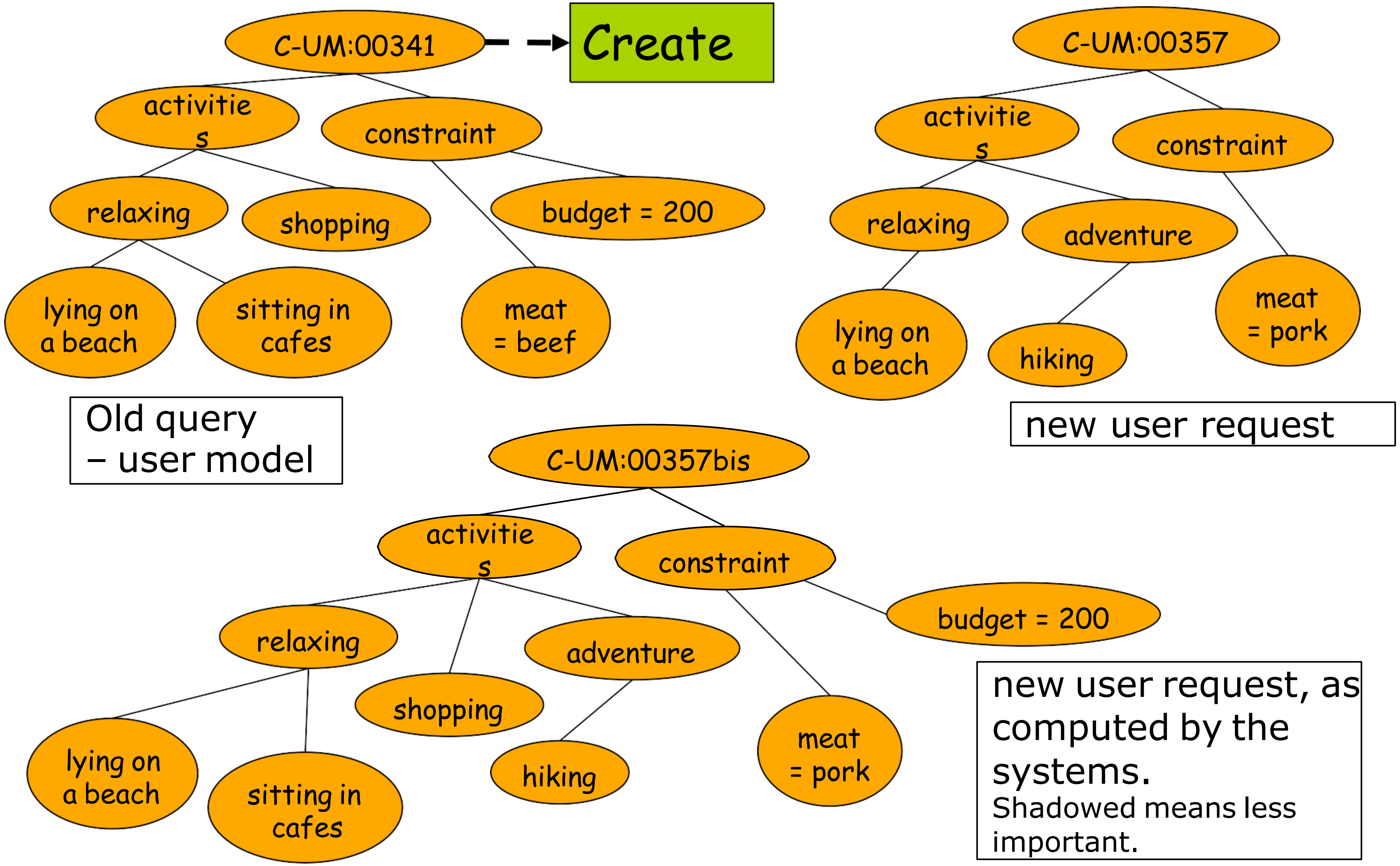
- The content (destination description) is exploited in the recommendation process
- A classical Content-Based method would have used a “simpler” content model ,e.g., keywords or TF-IDF
- Here a more complex **knowledge structure** – a tree of concepts – is used to model the product (and the query)

- The query is the user model and it is acquired every time the user asks for a new recommendation - (not exactly, more details later)

- Stress on ephemeral needs rather than building a persistent user model

- Typical in Knowledge-Based RS, they are more focused on ephemeral users – because Collaborative Filtering and Content-Based methods cannot cope with that users.

Knowledge-Based Recommender Systems



Learning User Profile: Query mining

Knowledge-Based Recommender Systems

Query Augmentation

- Personalization in search is not only "information **filtering**"
- **Query augmentation:** when a query is entered it can be compared against contextual and individual information to refine the query

- Ex1: If the user is searching for a restaurant and enter a keyword "Thai" then the query can be augmented to "Thai food"
- Ex2: If the query "Thai food" does not retrieve any restaurant the query can be refined to "Asian food"
- Ex3: If the query "Asian food" retrieves too many restaurant, and the user searched in the past for "Chinese" food the query can be refined to "Chinese food".

.

Query Augmentation in TripleHop

1. The current query is **compared** with **previous queries** of the **same user**

2. Preferences expressed in past (similar) queries are identified

3. A new query is built by **combining the short term preferences** contained in the query with the "**inferred**" preferences extracted from the persistent user model (past queries)

4. When the query is matched against an item (destination) if two destinations have the same degree of matching for the explicit preferences then the "inferred" preferences are used to break the tie

- This is another example of the cascade approach
- The two combined RS are based on the same knowledge but with two definitions of the user model.

What is Case Based Reasoning ?

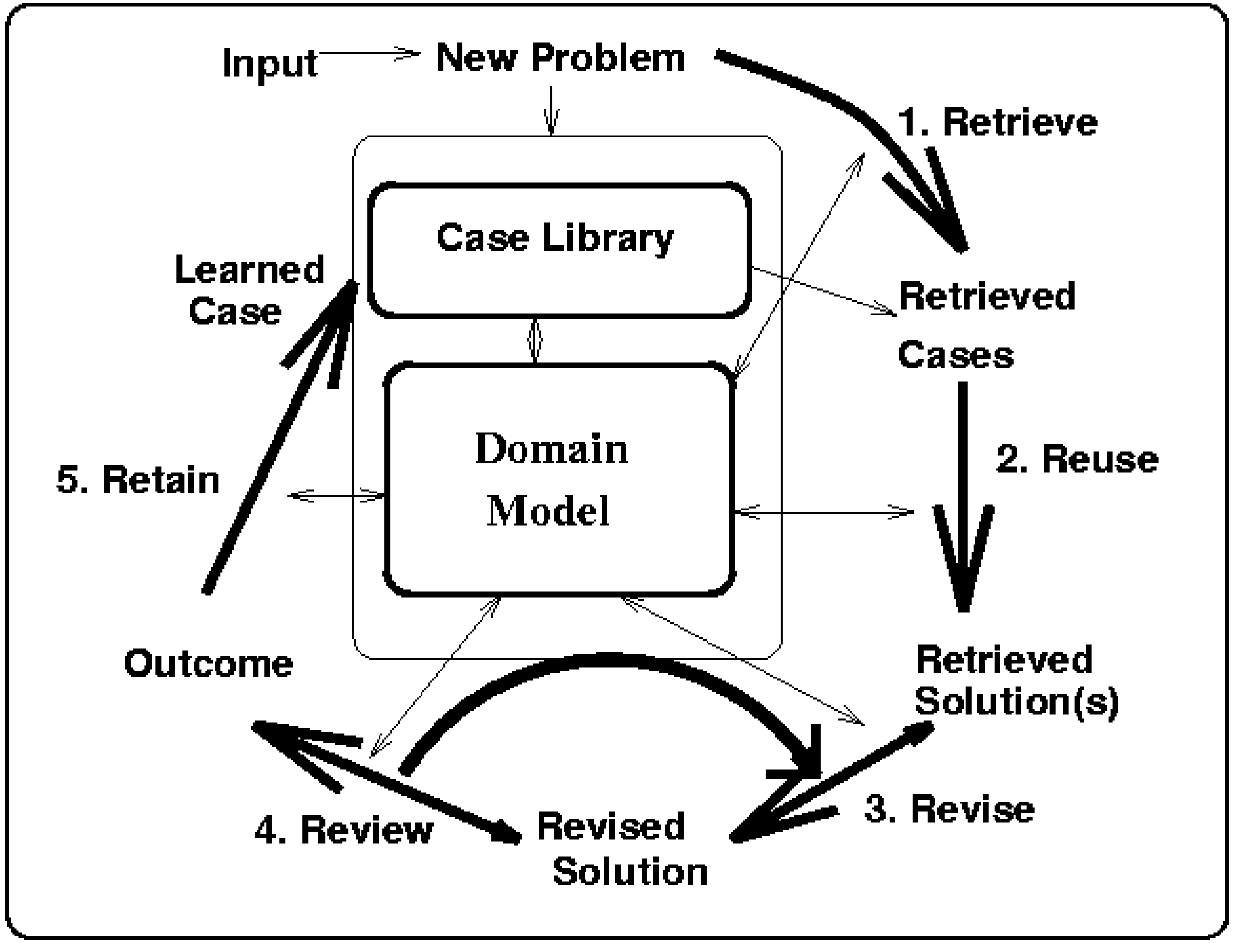
- **A case-based reasoner solves new problems by adapting solutions that were used to solve old problems** (Riesbeck & Shank 1989)

CBR problem solving process:

- Store previous experiences (cases) in memory to solve new problems
- Retrieve form the memory similar experience about similar situations
- Reuse the experience in the context of the new situation: complete or partial reuse, or adapt according to differences
- Store new experience in memory (learning)

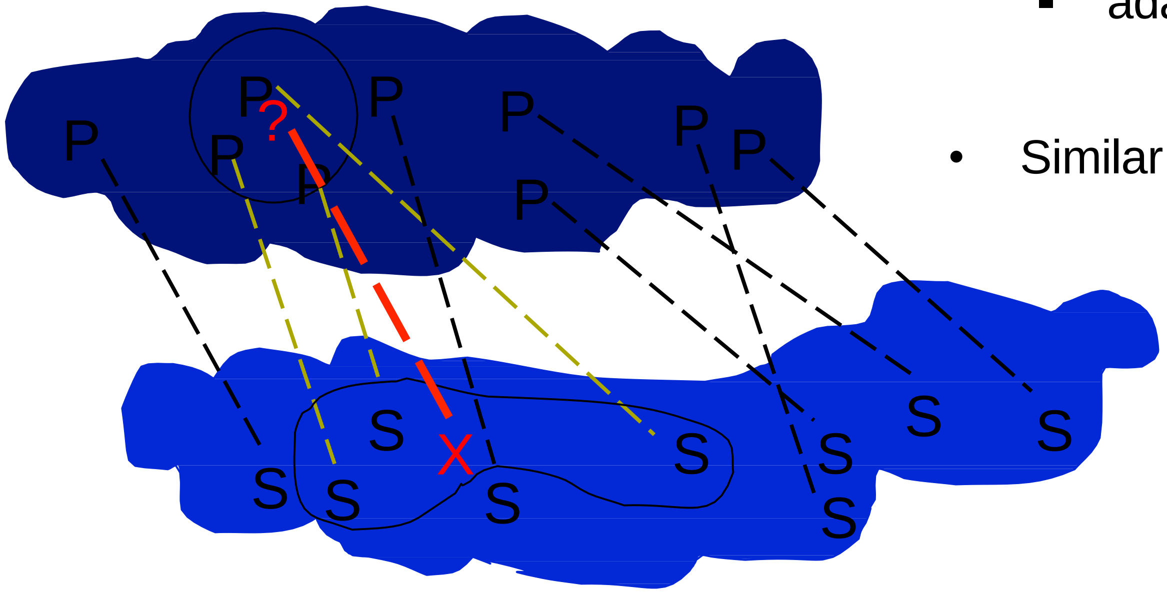
Knowledge-Based Recommender Systems

Case-Based Reasoning



Knowledge-Based Recommender Systems

CBR Assumption



- New problem can be solved by
  - retrieving similar problems
  - adapting retrieved solutions
- Similar problems have similar solutions

Knowledge-Based Recommender Systems

Examples of CBR

- Classification: “The patient`s ear problems are like this prototypical case of otitis media”
- Compiling solutions: “Patient N`s heart symptoms can be explained in the same way as previous patient D`s”
- Assessing values: My house is like the one that sold down the street for $250,000 but has a better view
- Justifying with precedents: “This Missouri case should be decided just like Roe v. Wade where the court held that a state`s limitations on abortion are illegal”
- Evaluating options: “If we attack Cuban/Russian missile installations, it would be just like Pearl Harbor”

Knowledge-Based Recommender Systems

Instance-based learning – Lazy Learning

- One way of solving tasks of approximating discrete or real valued target functions

- Have training examples: $(x_n, f(x_n))$, n=1,..

Key idea:

- Just store the training examples
- When a test example is given then find the closest matches
- Use the closest matches to guess the value of the target function on the test example.

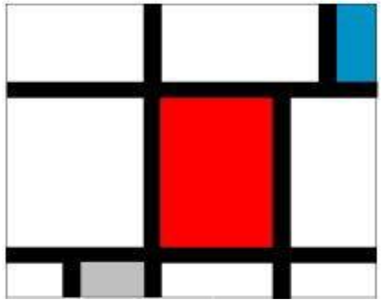
Knowledge-Based Recommender Systems

The distance between examples

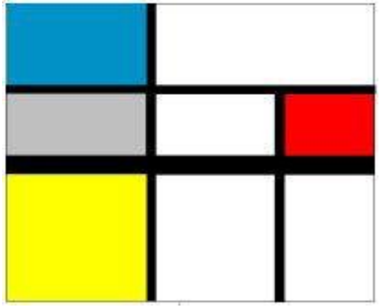
- We need a **measure of distance** (or similarity) in order to know who are the neighbors
- Assume that we have T attributes for the learning problem. Then one example point x has elements x_t , t=1,…,T
- The distance between two points x and y is often defined as the **Euclidean** distance:

$$d(x,y) = \sqrt{\sum_{t=1}^{T} [x_t - y_t]^2}$$

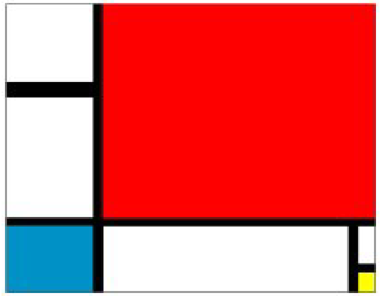
Knowledge-Based Recommender Systems



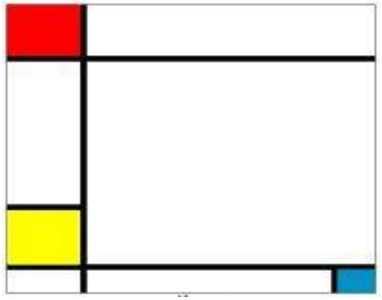
no



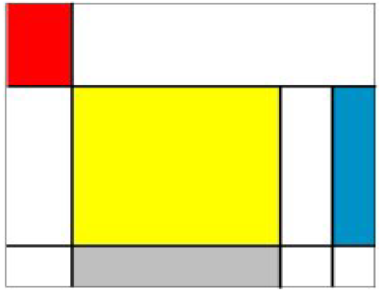
no



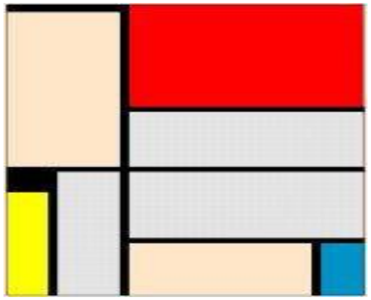
yes



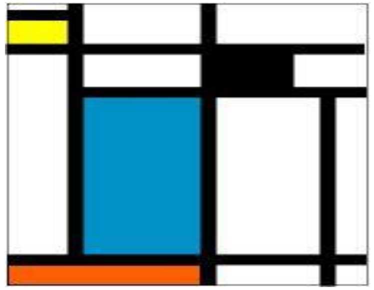
yes



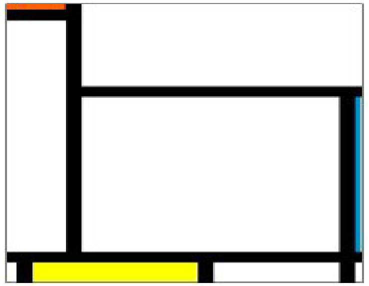
no



yes



no



?

Knowledge-Based Recommender Systems

Training data

Number	Lines	Line types	Rectangles	Colours	Mondrian?
1	6	1	10	4	*No*
2	4	2	8	5	*No*
3	5	2	7	4	Yes
4	5	1	8	4	Yes
5	5	1	10	5	*No*
6	6	1	8	6	Yes
7	7	1	14	5	*No*

Test instance

Number	Lines	Line types	Rectangles	Colours	Mondrian?
8	7	2	9	4	

References

Text Book:

“Recommender Systems, The Text Book by Charu C. Aggarwal, Springer 2016 Section 1 and Section 2.

Image Courtesy

http://www.mmds.org/mmds/v2.1/ch09-recsys1.pptx

https://www.researchgate.net/publication/287952023_Collaborative_Filtering_Recommender_Systems

http://cs229.stanford.edu/proj2014/Rahul%20Makhijani,%20Saleh%20Samaneh,%20Megh%20Mehta,%20Collaborative%20Filtering%20Recommender%20Systems.pdf

https://www.scribd.com/presentation/414445910/CS548S15-Showcase-Web-Mining

https://towardsdatascience.com/image-recommendation-engine-leverage-transfert-learning-ec9af32f5239

http://elico.rapid-i.com/recommender-extension.html
https://www.youtube.com/watch?v=h9gpufJFF-0

THANK YOU

Jyothi R.
Assistant Professor,
Department of Computer Science
jyothir@pes.edu