Knowledge-Based Recommender Systems

Francesco Ricci
eCommerce and Tourism Research Laboratory
Automated Reasoning Systems Division
ITC-irst
Trento – Italy
ricci@itc.it
http://ectrl.itc.it

Content

- Knowledge-based recommenders: definition and examples
- Case-Based Reasoning
- A recommender system exploiting a "simple" case model (the product is a case)
- Knowledge modeling and complex recommendation tasks
- A more complex example: the recommendation session is the case
- Ranking with two-fold similarity
- Empirical Evaluation
“Core” Recommendation Techniques

**U** is a set of users

**I** is a set of items/products

<table>
<thead>
<tr>
<th>Technique</th>
<th>Background</th>
<th>Input</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collaborative</td>
<td>Ratings from <strong>U</strong> of items in <strong>I</strong>.</td>
<td>Ratings from <strong>u</strong> of items in <strong>I</strong>.</td>
<td>Identify users in <strong>U</strong> similar to <strong>u</strong>, and extrapolate from their ratings of <strong>I</strong>.</td>
</tr>
<tr>
<td>Content-based</td>
<td>Features of items in <strong>I</strong></td>
<td><strong>u</strong>’s ratings of items in <strong>I</strong>.</td>
<td>Generate a classifier that fits <strong>u</strong>’s rating behavior and use it on <strong>I</strong>.</td>
</tr>
<tr>
<td>Demographic</td>
<td>Demographic information about <strong>U</strong> and their ratings of items in <strong>I</strong>.</td>
<td>Demographic information about <strong>u</strong>.</td>
<td>Identify users that are demographically similar to <strong>u</strong>, and extrapolate from their ratings of <strong>I</strong>.</td>
</tr>
<tr>
<td>Utility-based</td>
<td>Features of items in <strong>I</strong>.</td>
<td>A utility function over items in <strong>I</strong> that describes <strong>u</strong>’s preferences.</td>
<td>Apply the function to the items and determine <strong>i</strong>’s rank.</td>
</tr>
</tbody>
</table>

[Burke, 2002]

Knowledge Based Recommender

- 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
  - 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.**
ActiveBuyersGuide (Now)

digital camera product advisor

I want photos quality high enough... More Info
- 3.5" or more
- 5.0" or more
- No preference

My camera should fit inside... More Info
- Small
- Medium
- Large

I transfer cameras that have an Opinion.com rating of at least... More Info
- No preference

I want to spend... More Info
- From $ up to $

I want to zoom in on subjects across a... More Info
- Shoot from 3.5 ft away
- Shoot from 10 ft away
- No preference

My preferred brands... More Info
- Canon
- Nikon
- Sony

Preference: More Info
- Canon
- Nikon
- Sony

More Guidance

get results

ActiveBuyersGuide (Now)

camcorder product advisor

I want a camcorder that... More Info
- Occasional (casual)
- Regular
- Home video production
- No preference

My camcorder should fit inside... More Info
- Small
- Medium
- Large

I transfer camcorders that have an Opinion.com rating of at least... More Info
- No preference

I want to spend... More Info
- From $ up to $

I want to zoom in on subjects across a... More Info
- Close (1 ft away)
- Far (10 ft away)
- No preference

My camcorder should fit inside... More Info
- Small
- Medium
- Large

My preferred brands... More Info
- Canon
- Nikon
- Sony

Preference: More Info
- Canon
- Nikon
- Sony

More Guidance

get results

ActiveBuyersGuide (Now)

my3 player product advisor

My MP3 player (Diary Music Player) needs to be compatible with a... More Info
- Windows operating system
- Mac operating system

I want my MP3 player to hold... More Info
- A few songs (less than 120 MB)
- A few dozen songs (120 MB - 512 MB)
- Hundreds of songs (512 MB - 1 GB)
- Thousands of songs (1 GB or more)
- No preference

I prefer MP3 players that have an Opinion.com rating of at least... More Info
- No preference

I want to spend... More Info
- From $ up to $

My preferred brands... More Info
- Apple
- Sony
- Iriver

Preference: More Info
- Apple
- Sony
- Iriver

More Guidance

get results
Someplace Similar.

Now you can easily find a place that’s like a destination you’ve enjoyed before!

1. In which region is the destination you liked?
   - Europe

2. Choose the destination you liked, and we’ll find a similar spot.
   - Paris and Vicinity

[Let’s go!]

Someplace Similar.

Now pick a personality type that best describes YOU – this will help us find similar spots based on things you like.

- Cultural Creature
  - Loves Nothing more than culture – festivals, museums, local & international culture

- Beach Bum
  - Generally loves to hang out laid back with a glass of wine

- City Slicker
  - An urban dweller who goes where the action is. Clubs, parties, and the pulse of the city

- Anti Athlete
  - Always on the court or the course, whether it’s tennis, golf, or squash

- Shopping Shark
  - Stopped looking for your shoes when you stepped on the ground.

- Mountain Man
  - Wild, freeborn, and free of all social constraints and controls.

- Short Seeker
  - Always looking for that ticket out, now, or attention.

[Pick one and click!]

Operazione completata

Internet
Trip.com

Trip.com

Trip.com
Tell us more!

Give us a better idea about what you like. Feel free to skip any question, but the more you tell us, the better our recommendation will be.

Adventure Sports
- Any favorite adventure sports?
- Paragliding
- Rock-Climbing
- Whitewater Rafting

Relaxing
- Which of these do you enjoy?
- Enjoying Spa Treatments
- Sitting In Cafes
- Sitting In Parks
- Watching Sports

Save Your Preferences
Email Address: 

Search

We found 10 matches for you. To read more or book a vacation, please click on the destination name or the picture.

1. Monterey Bay - California
Like the Pacific Ocean that runs up and down the Monterey Bay and Big Sur coastline, it’s almost impossible to define and contain this area. There are many towns, each with a distinct flavor. Monterey, with its seal, sea otter, and whale titalite, ...more

- Dining in Cafes
- Hiking

2. Salem And The North Shore - Massachusetts
Imagine for a moment that you could disassemble yourself, slide Wienske or Star Trek style, into little bits, and then transplant yourself whole onto the pages of your favorite New England coffee table book. ...Right...more

- Dining in Cafes
- Hiking

3. Marin County - California
Often dubbed the "Van Nuys’ Backyard," Marin County is an area of recreational and geographic diversity. It is worth visiting for its location alone, as it is bordered by the Pacific Ocean, the Golden Gate Bridge, the San Francisco Bay, and Wine Country...

- Dining in Cafes
- Hiking
Matching in TripleHop

Example: TripleHop

- C-UM:00341
- activities
- constraint
- relaxing
- shopping
- budget = 200
- meat = beef
- lying on a beach
- sitting in cafes

Matching

Catalogue of Destinations

[Delgado and Davidson, 2002]

---

TripleHop and Content-Based RS

- The content (destination description) is exploited in the recommendation process
- A classical CB method would have used a “simple” content model: keywords or TF-IDF
- Here a more complex “knowledge structure” – a tree of concepts – is used to model the product
- The query is the user model and it is acquired every time the user asks for a new recommendation
  - Stress on ephemeral users rather than building a persistent user model.
- KB RS are more focused on ephemeral users – because CF and CB methods cannot cope with that users.
Learning User Profile

Query Augmentation

- Personalization in search does not reduce to “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

- The current query is compared with previous queries of the same user
- Preferences expressed in past (similar) queries are identified
- 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)

TripleHop : Discussion

- **Plus**
  - Some result in any case
  - Explanation for the matching
  - Use of profile information (shadowed)
- **Minus**
  - Decision process is not modelled (decision variables are too many and all used at the same time)
  - Fixed feature list (must fits all users)
  - Presentation is not personalized (standard text and multimedia content)
  - The “product” cannot be repackaged by the user
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)

[27]

Case-Based Reasoning

[Aha, 1998]
CBR Assumption

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

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”
CBR Knowledge Containers

- Cases
- Case representation language
- Retrieval knowledge
- Adaptation knowledge

Case representation language

- Contents
  - features and values of problem/solution
- Issues
  - more detail / structure = flexible reuse
  - less detail / structure = ease of encoding new cases
Retrieval knowledge

- Contents
  - features used to index cases
  - relative importance of features
  - what counts as "similar"
- Issues
  - “surface” vs “deep” similarity

Nearest Neighbour Retrieval

- Retrieve most similar
- k-nearest neighbour
  - k-NN
- Example
  - 1-NN
  - 5-NN
How do we measure similarity?

- Can be strictly numeric
  - weighted sum of similarities of features
  - “local similarities”
- May involve inference
  - reasoning about the similarity of items

Adaptation knowledge

- Contents
  - circumstances in which adaptation is needed
  - how to modify
- Issues
  - role of causal knowledge
    - “why the case works”
Learning

- Case-base
  - inserting new cases into case-base
  - updating contents of case-base to avoid mistakes
- Retrieval Knowledge
  - indexing knowledge
    - features used
    - new indexing knowledge
  - similarity knowledge
    - weighting
    - new similarity knowledge
- Adaptation knowledge

Example of CBR Recommender System

- Entree is a restaurant recommender system – it finds restaurants:
  1. in a new city similar to restaurants the user knows and likes
  2. or those matching some user goals (case features).
Partial Match

- In general, only a subset of the preferences will be matched in the recommended restaurant.

Nearest Neighbor
Recommendation in Entree

- The system first selects from the database the set of all restaurants that satisfy the largest number of logical constraints generated by considering the input features type and value
- If necessary, implicitly relaxes the lowest important constraints until some restaurants could be retrieved
- Sorts the retrieved cases using a similarity metric

Similarity in Entree

- This similarity metric assumes that the user goals, corresponding to the input features (or the features of the source case), could be sorted to reflect the importance of such goals from the user point of view
- Hence the global similarity metric (algorithm) sorts the products first with respect the most important goal and then iteratively with respect to the remaining goals (multi-level sort)
- **Attention:** *it does not works as a maximization of an Utility-Similarity defined as the sum of local utilities*
Utility and similarity

- If the user has some preferred values for the attributes, e.g., \( q_1, \ldots, q_m \), one can substitute to the value of the product a local similarity function \( \text{sim}(q_j, p_j) \):

\[
U(u_1, \ldots, u_n, q_1, \ldots, q_m, p_1, \ldots, p_n) = \sum_{j=1}^{m} u_j \text{Sim}(q_j, p_j)
\]

- A typical local similarity function is \( (1-|q_j - p_j|/\text{range}_i) \), where range is the difference between the max and min value of the attribute \( i \).

- A utility-based recommender becomes a similarity maximization recommender (or a nearest neighbor recommender).

- Utility and (simple) case-based reasoning approaches are strictly related (we’ll see in another lesson).

Example

<table>
<thead>
<tr>
<th>Restaurant</th>
<th>Price</th>
<th>Cusine</th>
<th>Atmosphere</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dolce</td>
<td>10</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>Gabbana</td>
<td>12</td>
<td>B</td>
<td>B</td>
</tr>
</tbody>
</table>

- If the user query \( q \) is: **price=9 AND cusine=B AND Atm=B**

- And the weights (importance) of the features is: 0.5 price, 0.3 Cusine, and 0.2 Atmosphere

- The Entrée will suggest Dolce first (and then Gabbana)

- A more traditional CBR system will suggest Gabbana because the similarities are (30 is the price range):

  - \( \text{Sim}(q, \text{Dolce}) = 0.5 * (29/30) + 0.3 * 0 + 0.2 * 0 = 0.48 \)
  
  - \( \text{Sim}(q, \text{Gabbana}) = 0.5 (27/30) + 0.3 *1 + 0.2 * 1 = 0.45 + 0.3 + 0.2 = 0.95 \)
Critiquing

- If the recommended restaurants satisfies the user then the interaction finishes
- But if the user is not satisfied, because of the values of some features of the proposed restaurant, then he can criticize them
- If for instance, the price is too high and the user is looking for something cheaper, then he/she can “tweak” the original request and provide a new input explicitly mentioning that the result must have a cheaper price
- This starts a new recommendation cycle and the criticized features is considered the most important user goal.

45

Input
Case Features: user's goals
Source Case: a known restaurant

New problem

6. Iterate
Tweaking: new user's goal

1. Retrieve
Logical query

Case library

Retrieved cases

2. Reuse

Domain model

Retrieved solutions

3. Revise
Ranking by similarity

4. Review

Revised solutions

Outcome

46
Entrée points

- **Plus**
  - Some results in any case
  - User may criticise the results (navigation in the info space)

- **Minus**
  - No explanation for the matching score
  - No comparison with alternative options
  - Fixed feature list (must fits all users)
  - No use of profile information

NutKing

- We shall now illustrate a Knowledge-based system that uses CBR as well
- It support a travel decision problem
  - Where to go
  - What to do
- Exploit a case model that describes the product to recommend but includes also some other "knowledge" useful to generate the recommendation.
Peculiarities of the Tourism industry

- I would like to escape from this ugly and tedious work life and relax for two weeks in a sunny place. I am fed up with these crowded and noisy places ... just the sand and the sea ... and some "adventure".

- I would like to bring my wife and my children on a holiday ... it should not be too expensive. I prefer mountainous places ... not too far from home. Children parks, easy paths and good cuisine are a must.

- I want to experience the contact with a completely different culture. I would like to be fascinated by the people and learn to look at my life in a totally different way.

Destinations and Location Search

? Destination?

? Where to stay, what to do?

or

? Destination?
Factors influencing Holiday Decision

**Internal to the tourist**
- Personality
- Motivators
- Disposable Income
- Health
- Family commitments
- Past experience
- Works commitments
- Hobbies and interests
- Knowledge of potential holidays
- Lifestyle
- Attitudes, opinions and perceptions

**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

Travel Planning

- **Problem:** Selecting a destination and related attraction/services - searching information available in Internet (e.g. in eTravel agencies, DMOS).
- **Complexity:**
  - **User variability:** needs and wants (constraints on products), personality, demographics, travel mean, travel party (constraints on traveler).
  - **Product complexity:** "destination" is a fuzzy concept. The same destination may have completely different fruition (e.g. religious, scenic, cultural, activity).
  - **Data and knowledge:** tourism information repository are complex and heterogeneous (data structure), distributed.
  - **User interface:** must ease the information navigation for all kinds of user on large repositories and supporting many functions (e.g. browsing, booking, comparing, filtering).
Travel Destination Choice Models

All Potential Destinations

Early Consideration Set

Awareness Unavailable and Unawareness Sets

Late Consideration Set

Inept and Inert Sets

Action Set

Inaction set

Alternatives for which information was sought but which were not selected

Final Destination

Choice Set Model
Crompton and Ankomah 1993

Travel Destination Choice Models

General Tourism Service Decision Process
Woodside and McDonald 1994
Travel Destination Choice Models

Decision Net Model
Fesenmaier and Jeng 2000

Important trip characteristics

<table>
<thead>
<tr>
<th>Trip characteristics</th>
<th>Research</th>
</tr>
</thead>
<tbody>
<tr>
<td>Travel purpose</td>
<td>Ankomah, Crompton, and Baker (1996), Um and Crompton (1991)</td>
</tr>
<tr>
<td>Travel party type/size</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Distance</td>
<td>Ankomah, Crompton, and Baker (1996), Kim and Fesenmaier (1990), Lo (1992), McKercher (1998)</td>
</tr>
<tr>
<td>Origin/Destination</td>
<td></td>
</tr>
<tr>
<td>Characteristics</td>
<td></td>
</tr>
<tr>
<td>Transportation mode</td>
<td>Cooper (1981), Tideswell and Faulkner (1999)</td>
</tr>
</tbody>
</table>
Important person’s characteristics

<table>
<thead>
<tr>
<th>Category</th>
<th>Variables</th>
<th>Research</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Education</td>
<td>Balcar and Pearce (1996), Mak and Moncur (1980)</td>
</tr>
<tr>
<td></td>
<td>Marital Status</td>
<td>Mak and Moncur (1980)</td>
</tr>
<tr>
<td></td>
<td>Attitude</td>
<td>Moutinho (1987), Um and Crompton (1990),</td>
</tr>
<tr>
<td></td>
<td>Involvement</td>
<td>Fesenmaier and Johnson (1989), Red and Crompton (1993)</td>
</tr>
</tbody>
</table>

Knowledge-Based System Perspective

- From a Knowledge-based system perspective all the previous characteristics (trip and traveler characteristics) should be:
  - **Modeled** – represented in a convenient way
  - **Exploited** in a reasoning process that starting from these and additional context-dependent information could derive the best recommendation
Travel Decision Theory

- Choice set models [Crompton and Ankomah, 1993]
- General Travel Models [Woodside and Lysonski, 1989]
- Decision net Models [Fesenmaier and Jeng, 2000]
- Multi-destination travel models [Lue et al., 1993]

- Personal characteristics: socio-economics, needs and wants, psychological/cognitive traits
- Travel characteristics: travel purpose, travel party size, length of travel, distance, transportation mode.

How to incorporate such conceptual approaches into a human centric travel planning recommender system?

Requirements and Issues

- Recommendation Process
  - Recommendation requires information search and discovery – not only filtering
  - Human/Computer dialogues should be supported – e.g. user criticizes a suggested product or refine a query definition

- Input/Output
  - Products and services may have complex structures
  - The final recommendation is a bundling of elementary components
  - Allow system bootstrapping without an initial memory of rates interactions
  - Generalize the definition of rates (implicit rates)

- Users
  - Both short term (goal oriented) preferences and long term (stable) preferences must influence the recommendation
  - Unregistered users should be allowed to get recommendations
  - Account for user variability in preferred decision style
  - Users needs and wants structure/language may not match those of the products.
Features of Complex Products

- Aggregation of more elementary components
- Multiple decisions must be taken (which destination, which mode of travel, which type of accommodation, how long, when, package or independent, which operator, ...)
- Can be viewed as a configuration problem
- Products change with time – “cold start” is not only at the beginning!
- Usually the product has a high (monetary) value
- Complexity increase the risk to take the wrong decision
- Difficult to standardize and have a unique approach – system design becomes crucial

Classical CBR Recommendation

- **Recommending by searching travel plans (or elementary travel services) similar to a partially defined one (some features).**
- **Pros:**
  - relatively simple to implement and easy to explain
  - takes into account user needs and wants
- **Cons:**
  - Bootstrap requires case to be "manually' built
  - How to recommend single travel components (e.g. an hotel, event, etc.) – each of them requires a new case base
  - How to exploit multiple case bases and learn to bundle “solutions” coming from multiple case bases
  - The user query (by similarity) tends to be not very selective, i.e., many roughly similar plans are retrieved
  - Default approaches are not "conversational", i.e., the initial query cannot be refined (i.e. become more specific or more generic).
### Travel Companies
- With family

### Transport
- How will you travel?
  - Car

### Accommodation
- What kind of accommodation do you want?
  - Hotel
- What's your daily budget for accommodation?
  - Between 20 and 40 €

### Departure
- Where are you from?
  - Italy
- When do you want to travel?
  - August
- How long do you want to stay?
  - One week
- Have you ever visited Trentino?
  - No

### Activities
- What would you like to do on this trip?
  - Sports
  - Adventure
  - Reising
  - Art & Culture
  - Wine and Food
  - Environment and Landscape
  - Fitness and Wellness
Query Tightening

24 results

I found 24 results that matched your request. Below we suggest ways to modify your request and receive more refined results.

- Add "cost" to your query.
- Add "exclusive" to your query.
- Add "Trip" to your query.

Skip the refinement ▼ Get all results

Update research

Sorry, we don't have anything to satisfy your requirements. You can change your request by:

- Remove and Get results
- Remove and Get results
- Remove and Get results
- Remove and Get results
Our “case” for CBR

- Problem = recommend a set of tourism related products and build a travel plan
- Cases = All the recommended travel plans that users have built using the system (how they were built and what they contain)
- Retrieval = search in the memory travel plans built during “similar” recommendation sessions
- Reuse
  1. extract from previous travel plans elementary components (items) and use them to build a new plan
  2. rank items found in the catalogues
Case/Session Model

Collaborative Features
- travel-party = single, budget= 20-40, sport=T, eno-gastronomy=T

Content Queries
- DESTINATION where rockclimbing=T AND hiking=T
- HOTEL where category=3 AND parking=T AND cost<40

Collaborative Component 2: selected products
- (Kitzbühel, True, True,...)
- (Hotel Schwarzer, 3, True,...)

Travel Plan and Interaction Session Model

July in Fiemme Valley

Collaborative Component 1: travel wish
- clf
  (family, bdg_medium,7,Hotel)

Queries on content attributes
- cnq
  (Golfing=True AND Nightlife=True)
  (category=3 AND Health=True)

Collaborative Component 2: selected products
- Travel bag
- rating
Item Ranking

1. Search the catalogue

2. Search Similar Cases

3. Output Reference Set

4. Sort locations by similarity to locations in reference cases

Interactive query management

Locations from Catalogue

Case Base

Input Q

Travel components

Current Case

Ranked Items

Output

Single Item Recommendation Process

RecommendItem(cnq)

IQM

Catalog

RecEngine

Ranked products

SimCases(clf)

Case Base

query(q)

products or query refinement

(1) EvaluateQuery(q)

(2) Rank(clf, products)

ranked products or query refinement
Rank using Two-Fold Similarity

- Given the current session case $c$ and a set of retrieved products $R$ (using the interactive query management facility - IQM)
  1. retrieve 10 cases ($c_1, ..., c_{10}$) from the repository of stored cases (recommendation sessions managed by the system) that are most similar to $c$ with respect to the collaborative features
  2. extract products ($p_1, ..., p_{10}$) from cases ($c_1, ..., c_{10}$) of the same type as those in $R$
  3. For each product $r$ in $R$ compute the Score($r$) as the maximal value of the product of a) the similarity of $r$ with $p_i$, b) the similarity of the current case $c$ and the retrieved case $c_i$ containing $p_i$
  4. sort and display products in $R$ according to the Score($p$).
**Ranking: Graphical Representation**

![Graphical Representation Diagram]

**Example: Scoring Two Destinations**

Destinations matching the user's query:

- **D1**
- **D2**

Current case CC:

- **C1**
- **C2**

Similar cases in the case base:

- **CD1**
- **CD2**

Score calculation:

\[
\text{Score}(D_i) = \max_j \{\text{Sim}(CC, C_j) \times \text{Sim}(D_i, CD_j)\}
\]

Score calculation for **D1**:

\[
\text{Sim}(CC, C1) = 0.2, \quad \text{Sim}(D1, CD1) = 0.4
\]

\[
\text{Score}(D1) = \max\{0.2 \times 0.4, 0.6 \times 0.7\} = 0.42
\]

Score calculation for **D2**:

\[
\text{Sim}(CC, C2) = 0.6, \quad \text{Sim}(D2, CD1) = 0.5
\]

\[
\text{Sim}(D2, CD2) = 0.3
\]

\[
\text{Score}(D2) = \max\{0.2 \times 0.5, 0.6 \times 0.3\} = 0.18
\]
**Bipartite Graph**

User-to-user CF exploits user-to-user similarities

Item-to-item CF exploits item-to-item similarities

Two-fold similarity exploits both similarities

**Case/Session Model**

**Case**

Collaborative Features

travel-party = single, budget= 20–40, sport=T, eno-gastronomy=T

**Content Queries**

DESTINATION where rockclimbing=T AND hiking=T

HOTEL where category=3 AND parking=T AND cost<40

**Cart**

Destinations

name=Canazei, rockclimbing=T, hiking=T, museum=T, canoeing=F

name=Molveno, rockclimbing=T, hiking=T, museum=F, canoeing=T

Accommodations

type=hotel, name=Irma, category=3, cost = 35, parking=T
Tree-based Case Representation

- A case is a rooted tree and each node has a:
  - **node-type**: similarity between two nodes in two cases is defined only for nodes with the same node-type
  - **metric-type**: node content structure - how to measure the node similarity with another node in a second case

![Tree-based Case Representation Diagram]

Item Representation

**TRAVELDESTINATION** = (X₁, X₂, X₃, X₄)

<table>
<thead>
<tr>
<th>Node Type</th>
<th>Metric Type</th>
<th>Example: Canazei</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCATION</td>
<td>Set of hierarchical related symbols</td>
<td>Country=ITALY, Region=TRENTINO, TouristArea=FASSA, Village=CANAZEI</td>
</tr>
<tr>
<td>INTERESTS</td>
<td>Array of booleans</td>
<td>Hiking=1, Trekking=1, Biking=1</td>
</tr>
<tr>
<td>ALTITUDE</td>
<td>Numeric</td>
<td>1400</td>
</tr>
<tr>
<td>LOCTYPE</td>
<td>Array of booleans</td>
<td>Urban=0, Mountain=1, Riverside=0</td>
</tr>
</tbody>
</table>

![Item Representation Diagram]
Item Query Language

- For querying purposes items $x$ are represented as simple vector features $x=(x_1, \ldots, x_n)$

\[ X_1 = (\text{Italy, Trentino, Fassa, Canazei}) \]

\[ X_2 = (1,1,1) \]

\[ X_3 = 1400 \]

\[ X_4 = (0, 1, 0) \]

(IItaly, Trentino, Fassa, Canazei, 1, 1, 1, 1400, 0, 1, 0)

- A query is a conjunction of constraints over features: 
  \[ q = c_1 \land c_2 \land \ldots \land c_m \] 
  where $m \leq n$ and 

  \[
  c_k = \begin{cases} 
  x_k = \text{true} & \text{if } x_k \text{ is boolean} \\
  x_k = v & \text{if } x_k \text{ is nominal} \\
  l \leq x_k \leq u & \text{if } x_k \text{ is numerical}
  \end{cases}
  \]

Scoring

- A collection of case sessions $s_i, i=1, \ldots, n$ and a collection of items/products $p_j, j=1, \ldots, m$

- A $n \times n$ sessions similarity matrix $S=\{s_{ij}\}$ and a $m \times m$ items/products similarity matrix $P=\{p_{ij}\}$

- A $n \times m$ session vs. product incidence matrix $A=\{a_{ij}\}$, where $a_{ij}=1$ ($a_{ij}=0$) if session $i$ does (not) include product $j$

  \[
  \text{Score}(s_i, p_j) = \text{MAX}_{k, l}\{s_{ik} a_{kl} p_{lj}\}
  \]

- A product $p_j$ gets a high score if it is very similar ($p_{lj}$ close to 1) to a product $p_l$ that is contained in a session $s_k$ ($a_{kl}=1$) that is very similar to the target session ($s_{ik}$ close to 1)

- A particular product can be scored (high) even if it is not already present in other sessions/cases, provided that it is similar to other products contained in other similar sessions.
**Item Similarity**

If X and Y are two items with same node-type, the similarity is defined as

\[
d(X,Y) = \left(\frac{1}{\sum_i w_i}\right)^{1/2} \left[\sum_i w_i d_i(X_i, Y_i)^2\right]^{1/2}
\]

where \(0 \leq w_i \leq 1\), and \(i=1..n\) (number of features).

\[
d_i(X_i, Y_i) = \begin{cases} 
1 & \text{if } X_i \text{ or } Y_i \text{ are unknown} \\
\text{overlap}(X_i, Y_i) & \text{if } X_i \text{ is symbolic} \\
|X_i - Y_i|/\text{range}_i & \text{if } X_i \text{ is finite integer or real} \\
\text{Jaccard}(X_i, Y_i) & \text{if } X_i \text{ is an array of Boolean} \\
\text{Hierarchical}(X_i, Y_i) & \text{if } X_i \text{ is a hierarchy} \\
\text{Modulo}(X_i, Y_i) & \text{if } X_i \text{ is a circular feature (month)} \\
\text{Date}(X_i, Y_i) & \text{if } X_i \text{ is a date}
\end{cases}
\]

Sim(X,Y) = 1 - d(X,Y) or Sim(X,Y) = exp(- d(X,Y))

---

**Local Metric Catalogue**

- **Jaccard**: distances among two group of Boolean features
- **Hierarchical**: distances among two nodes whose children leaves are linked by a hierarchical relationships (Location as Country, Region, TourismArea, Village)
- **Modulo**: distances among circular features (e.g. months)
- **Date**: distance among dates
- **Symbolic**: distance among symbolic features
- **Numeric**: distance among numeric features
Metric Catalogue: Jaccard

This metric is used to calculate the distance between two groups of Boolean features. Unknown values must be converted to zeroes before.

\[ d_i(x_i, y_i) = \frac{N_{01} + N_{10}}{N_{01} + N_{10} + N_{11}} \]

\( N_{01} \) is the number of entries that are 0 in the first array and 1 in the second.

Similar definitions for the others.

| \( x_i \) | 1 | 0 | 0 | 0 | 0 | 0 |
| \( y_i \) | 0 | 1 | 0 | 0 | 0 | 0 |

\[ \text{dist} = \frac{2}{2} = 1 \]

| \( x_i \) | 1 | 0 | 1 | 1 | 1 | 1 |
| \( y_i \) | 0 | 1 | 1 | 1 | 1 | 1 |

\[ \text{dist} = \frac{2}{6} = 0.3 \]

---

Metric Catalogue: HierSet

Used for distances between two groups of Hierarchical features.

The distance at the lowest level of abstraction where the two objects are different.

\( \text{hierset}(x, y) = \min(d_i(x_i, y_i)) \)

\[ d_i(x_i, y_i) = \begin{cases} 1 & \text{if } x_i \neq y_i \\ v_i & \text{if } x_i = y_i \end{cases} \]

<table>
<thead>
<tr>
<th>feature</th>
<th>( x_i )</th>
<th>( y_i )</th>
<th>equality</th>
<th>( v_i )</th>
<th>( d_i(x_i, y_i) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>country</td>
<td>Italy</td>
<td>Italy</td>
<td>=</td>
<td>0.7</td>
<td>0.7</td>
</tr>
<tr>
<td>region</td>
<td>Trentino</td>
<td>Trentino</td>
<td>=</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>tourisarea</td>
<td>Val di Fiemme</td>
<td>Val di Fiemme</td>
<td>=</td>
<td>0.3</td>
<td>0.3</td>
</tr>
<tr>
<td>village</td>
<td>Caiazzo</td>
<td>Predazzo</td>
<td>\neq</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

min dist: 0.3
Metric Catalogue: Modulo

- The modulo metric is used for computing the distance between two “circular” feature values like two months (mod = 12)
- If x and y are two month values, then their distance is:
  - Min{(x-y)mod(12), (y-x)mod(12)}/12
- Example:
  - x=1, y=11: Min{-10mod(12), 10mod(12)}/12 = Min{2, 10}/12 = 2/12
  - The opposite of 10 mod(12) is the number that summed to 10 produces 0 mod(12), hence it is 2.

Metric Catalogue: Symbolic

\[ d_i(x_i, y_i) = \begin{cases} 
0 & \text{if } x_i = y_i \\
1 & \text{if } x_i \neq y_i \\
1 & \text{if } x_i \text{ or } y_i = \text{null} 
\end{cases} \]

- The distance between a known value and an unknown value is always 1
- This holds also for the other local distances
Metric Catalogue: Numeric

\[ d_i(x_i, y_i) = \begin{cases} \frac{|x_i - y_i|}{4\sigma_i} & \text{if } \frac{|x_i - y_i|}{4\sigma_i} > 1 \\ 1 & \text{if } x_i \text{ or } y_i = \text{null} \end{cases} \]

- \( \sigma_i \) is the standard deviation. We are assuming the Chebyshev’s rule
  - Regardless of how the data are distributed, the percentage of observations that are contained within a distance of \( k \) standard deviations of the mean is at least \((1 - \frac{1}{k^2})\)100%
  - Empirical rule: approximately 95% of the data points will have a value within 2 standard deviation of the mean
- A range of 4 \( \sigma_i \) contains almost all the data

Similarity

\[ \text{Sim}(C, C') = \exp(-d(C, C')) \]
**Item Similarity Example**

\[
X_1 = (I, TN, Fassa, Canazei) \\
X_2 = (1, 1, 1) \\
X_3 = 1400 \\
X_4 = (0, 1, 0)
\]

\[
Y_1 = (I, TN, Fassa, ?) \\
Y_2 = (1, 0, 1) \\
Y_3 = 1200 \\
Y_4 = (1, 1, 0)
\]

\[
Sim(dest_1, dest_2) = \exp\left(-\frac{1}{\sqrt{4}}\sqrt{d_1(X_1, Y_1)^2 + \cdots + d_4(X_4, Y_4)^2}\right)
\]

\[
= \exp\left(-\frac{1}{\sqrt{4}}\sqrt{(1/4)^2 + (1/3)^2 + ((1400-1200)/2000)^2 + (1/2)^2}\right)
\]

\[
= \exp\left(-\frac{1}{\sqrt{4}}\sqrt{0.4361}\right) = \exp(-0.3301) = 0.7188
\]

**Case Distance**

- **nt**: cart
- **nt**: vector
- **nt**: destination
- **nt**: set
- **nt**: location
- **nt**: hierarchical

\[
\begin{align*}
\text{c1} & \quad \text{cart1} \\
\text{clf1} & \quad \text{cnq1} \\
\text{c1} & \quad \text{dests1} \\
\text{acts1} & \quad \text{dest1} \\
\text{X}_1 & \quad \text{X}_2 \\
\text{X}_3 & \quad \text{X}_4 \\
\text{c2} & \quad \text{cart2} \\
\text{clf12} & \quad \text{cnq2} \\
\text{c2} & \quad \text{dests2} \\
\text{acts2} & \quad \text{dest2} \\
\text{Y}_1 & \quad \text{Y}_2 \\
\text{Y}_3 & \quad \text{Y}_4 \\
\end{align*}
\]
Case Distance

\[ d(c_1, c_2) = \frac{1}{\sum_{i} W_i} \sqrt{W_1 d(cart_1/cart_2)^2 + W_2 d(clf_1/clf_2)^2 + W_3 d(cnq_1/cnq_2)^2} \]

\[ d(cart_1/cart_2) = \sqrt{d(dests_1/dests_2)^2 + d(clf_1/clf_2)^2 + d(cnq_1/cnq_2)^2} \]
Empirical Evaluation

- Two groups of users
  - **NutKing-**: without the scoring mechanism (cases are not used) - 19 users
  - **NutKing+**: the scoring mechanism implemented as described before - 16 users – the case base contained 35 cases

- We compared the: **position of the selected item in the ranking computed by the system**

- If the user puts in the cart the products that are highly ranked this means
  - The ranking is correct, i.e., respect the true user ranking (accuracy)
  - The user is going to make less effort in the search (cost)

[Ricci et al., 2003]
Empirical Evaluation

- The two systems have exactly the same graphical interface
- We performed a between subject test
  - A user is assigned to only one of the two systems
  - No user can recognize that there is a system A and B (e.g., one more simple and one more advanced)
- We measured a difference that is significant for a t-test
  - NutKing+: average position = \(2.2 \pm 2.9\)
  - NutKing−: average position = \(3.2 \pm 4.8\)

Objective Measures

<table>
<thead>
<tr>
<th></th>
<th>NutKing−</th>
<th>NutKing+</th>
</tr>
</thead>
<tbody>
<tr>
<td>Queries issued by the user</td>
<td>20.1±19.1</td>
<td>(13.4 \pm 9.3)</td>
</tr>
<tr>
<td>General travel wishes provided</td>
<td>12.3±1.6</td>
<td>11.5±2.0</td>
</tr>
<tr>
<td>Constraints per query</td>
<td>4.7±1.2</td>
<td>4.4±1.0</td>
</tr>
<tr>
<td>Results per query</td>
<td>42.0±61.2</td>
<td>(9.8 \pm 14.3)</td>
</tr>
<tr>
<td>Pages displayed</td>
<td>93.4±44.3</td>
<td>(71.3 \pm 35.4)</td>
</tr>
<tr>
<td>Items in the final travel plan</td>
<td>5.8±3.9</td>
<td>4.1±3.4</td>
</tr>
<tr>
<td>Session duration</td>
<td>28.5±9.5</td>
<td>27.3±13.0</td>
</tr>
<tr>
<td>Calls to query relaxation</td>
<td>n.a.</td>
<td>6.3±3.6</td>
</tr>
<tr>
<td>User accepted relax suggestion</td>
<td>n.a.</td>
<td>2.8±2.1</td>
</tr>
<tr>
<td>Calls to query tightening</td>
<td>n.a.</td>
<td>2.1±2.5</td>
</tr>
<tr>
<td>User accepted tightening sugg.</td>
<td>n.a.</td>
<td>0.6±0.9</td>
</tr>
<tr>
<td>Position of the selected item</td>
<td>3.2±4.8</td>
<td>(2.2 \pm 2.9)</td>
</tr>
</tbody>
</table>

Bold face means significantly different (t-test, \(p<.05\)).
Case-Based Cycle Revised

1. Retrieve

2. Reuse

3. Revise

4. Review

5. Retain

6. Iterate

Input (User Input)

New Problem

Retrieved Cases

Retrieved Solution(s)

Outcome

Input

Product features: Content
Query

Source Case: Collaborative
Features

Case library

Domain model

Product catalogue

Result set

1. Retrieve Similarity

2. Reuse

Retrieved cases

Retrieved solutions

3. Revise Ranking

4. Review

Edit case

5. Retain

Outcome

 Learned Case

Case Base (experiences)
Travel Completion in DieToRecs

- Design and development of a new algorithm for travel completion (cross selling)
- Personalized suggestions of additional products which fit well in the current travel (considering the products already selected by the user)
User Model in NutKing

- The user model comprises three components
  - Persistent information about the user (age, sex, occupation) – never used in the recommendation process (but useful for other purposes)
  - Information explicitly given at the beginning of the interaction – collaborative features
  - User actions during the interaction
    - Queries made
    - Product selected (if any)
What is a User “Model”

- Is the current case: short term preferences – built even for ephemeral (not registered users)
- Is the user profile: socio-demographic data
- Is the set of user’s previous cases: used to derive long term preferences (in another application for a mobile phone)
- Is the set of cases similar to the current one: reused in the CBR problem solving loop to rank user selections or complete a partial travel plan (cross selling)

Instance-based approach – lazy approach – no abstraction is built

NutKing as a “Hybrid” System

- In general NutKing incorporates idea from IR, content-based filtering and social filtering
- The IQM functionality is a content-based approach (but does not build a classifier for each user)
- Exploits a “Cascade” approach: first the IQM filters the catalogue, then the double similarity ranking operates
- Exploits a “Meta-Level” approach: the cases built by the Knowledge based (content based) system are used to implement a sort of collaboration via case description
- Extends CF since both new users and new products can be ranked
  - It uses the session model rather than a user model
  - Similar products are considered to be “almost equal”
Software Infrastructure

Web browser

Mobile J2ME

Application Server

Trip@device

Recommendation Services

STRUTS

XML/JDOM

JDBC

J2EE

Oracle 9i

Recommendation Services Structure

GUI Prototype

Recommendations ↔ and user’s status

Recommendation Framework

CaseEngine current case

Recommendation Engine

Metrics similarity functions

XMLMediator

Products and cases

Catalog1 Catalog2 CatalogN Case Base
Conclusions

- Knowledge-based systems requires a big effort in term of system design
- KB systems uses a lot of techniques (not simple content-based or collaborative based filtering)
- Many KB recommender systems are rooted in Case-Based Reasoning
- Other KB systems are rule-based (in commercial products): let the marketer to design the personalization
- Similarity of complex data objects is required

Questions

- What are the main differences between a CF recommender system and a KB RS (such as activebuyers.com)?
- What is the basic rationale of a CBR recommender system?
- What are the advantages and disadvantages of the double-similarity ranking method?
- KB recommender try to incorporate the findings derived by researchers that have analyzed the user decision processes (without system support)?
- Could a CF or Content-based recommender system incorporate such decision models?