Part 15: Knowledge-Based Recommender Systems

Francesco Ricci
Content

- Knowledge-based recommenders: definition and examples
- Case-Based Reasoning
- Instance-Based Learning
- A recommender system exploiting a “simple” case model (the product is a case)
- A more complex CBR recommender system for travel planning
### “Core” Recommendation Techniques

<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 $U$ of items in $I$.</td>
<td>Ratings from $u$ of items in $I$.</td>
<td>Identify users in $U$ similar to $u$, and extrapolate from their ratings of $i$.</td>
</tr>
<tr>
<td>Content-based</td>
<td>Features of items in $I$</td>
<td>$u$’s ratings of items in $I$</td>
<td>Generate a classifier that fits $u$’s rating behavior and use it on $i$.</td>
</tr>
<tr>
<td>Demographic</td>
<td>Demographic information about $U$ and their ratings of items in $I$.</td>
<td>Demographic information about $u$.</td>
<td>Identify users that are demographically similar to $u$, and extrapolate from their ratings of $i$.</td>
</tr>
<tr>
<td>Utility-based</td>
<td>Features of items in $I$.</td>
<td>A utility function over items in $I$ that describes $u$’s preferences.</td>
<td>Apply the function to the items and determine $i$’s rank.</td>
</tr>
<tr>
<td>Knowledge-based</td>
<td>Features of items in $I$. Knowledge of how these items meet a user’s needs.</td>
<td>A description of $u$’s needs or interests.</td>
<td>Infer a match between $i$ and $u$’s need.</td>
</tr>
</tbody>
</table>

[Burke, 2007]
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, 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.**
Wizard: My Product Advisor

Now you can:
- Answer more questions that are important to you.
- See recommended cameras based on your preferences so far.
- Review what you have done or start over.

Possible user's requests

The system decides what the wizard says.
Someplace Similar.
Now you can easily find a place that's like a destination you've enjoyed before!

Q1. In which region is the destination you liked?

Europe

Q2. 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.

- **Culture Creature**
  Loves everything cultural - theater, shows, museums... local & historical culture tool

- **Beach Bum**
  Somebody has to lay around on the beach with little umbrellas pitched in their drinks.

- **Trail Trekker**
  If it’s outdoors - you’re there.
  Hiking, walking... parks, forests, mountains.

- **Sight Seeker**
  Always looking for that landmark, event, or attraction.

- **City Slicker**
  An urban creature who goes where the action is. Clubs, people... love the pulse of the city.

- **Avid Athlete**
  Always on the court or the course... always in the game... whatever game it is.

- **Shopping Shark**
  Stopped looking for a cure for your shop-aholism?

- **Winter Warrior**
  Will work for lift ticket. Can become quite abominable if there’s no snow on the ground.

{pick one and click!}
If you liked Paris and Vicinity, you'll probably like these destinations as well:

<table>
<thead>
<tr>
<th>MATCH</th>
<th>DESTINATION</th>
<th>FIND OUT MORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>88%</td>
<td>New York City, NY</td>
<td>more</td>
</tr>
<tr>
<td>87%</td>
<td>Berlin</td>
<td>more</td>
</tr>
<tr>
<td>87%</td>
<td>London</td>
<td>more</td>
</tr>
<tr>
<td>85%</td>
<td>Greater Montreal, QC</td>
<td>more</td>
</tr>
<tr>
<td>85%</td>
<td>Beijing</td>
<td>more</td>
</tr>
<tr>
<td>83%</td>
<td>Washington D.C.</td>
<td>more</td>
</tr>
<tr>
<td>83%</td>
<td>Philadelphia and Lehigh Valley, PA</td>
<td>more</td>
</tr>
<tr>
<td>83%</td>
<td>Chicagoland Region, IL</td>
<td>more</td>
</tr>
<tr>
<td>83%</td>
<td>Hesse (Frankfurt and Vicinity)</td>
<td>more</td>
</tr>
<tr>
<td>82%</td>
<td>Greater Boston, MA</td>
<td>more</td>
</tr>
</tbody>
</table>

*Want to try Someplace Similar with a different destination? Click here.*
About New York City, NY

Recommended for: Culture Creature
Cost: $364-793

Overall Score: 90%

Overview
Where to, Mack? Central Park? You got it. First time to the Big Apple? Well, that's the Manhattan skyline over there - $24 in glass beads. The deal of the last millennium, I call it. Then, of course, we have Queens, the Bronx, Staten Island, and Brooklyn, where yours truly was born. In these five boroughs you'll find more landmarks, history, museums, restaurants, shopping, and people than I got problems. Can I name one of each? With my eyes closed. Relax, Mack! It's just a figure of speech. The Empire State Building, The American Museum of Natural History, the Metropolitan Museum of Art, the Colosseum, Eiffel Tower, and the Colosseum. He lives on 36th and Amsterdam. I see you like to be entertained. Well, for you, we got theater, nightlife (and I mean all night, Mack), music, and sports. Where? You don't get out much, huh? Ever heard of Broadway, Times Square, Lincoln Center, and the Bronx Bombers? The Yankees. Right - I see your moods are kickin' in. We also got the NFL, NHL, NBA, tennis in Flushing, and ping-pong in Chinatown. Say what? You like multiculturalism? You mean who lives here, right? EVERYBODY! Name a country and you have a little piece of New York. OK, my friend, we're here. They'll be 80 kids. It seems expensive? Welcome to New York, sweetheart!

Principal Cities
New York City
Trip Coach

People are as different as the trips they take. That's why Trip Coach finds destinations for you based on your travel interests. Select a personality or create your own, and we'll find destinations that are great for you.

Select the personality below that best describes you.

- **Winter Warrior**
  - All you need on your trip is snow, skiing, snow boarding, and hanging out at the lodge mark your final destination.

- **Sports Enthusiast**
  - Whether spectator or participant, your ideal trip involves anything sports-related: golf, tennis, baseball, football, and everything in between.

- **Sight Seeker**
  - You revel in trips that keep you busy searching for the next tour, attraction, or landmark.

- **Seasoned Shopper**
  - Your motto is "shop 'til you drop." For you, traveling is all about finding the best shops and bargains in town.

- **Outdoor Adventurer**
  - The great outdoors and all that goes with it: hiking, biking, kayaking, canoeing, skiing, exploring - is your idea of a perfect getaway.

- **Family Traveler**
  - From amusement parks to festivals to outdoor fun, you love to travel with your children, or you're just a kid at heart. Either way, your trip is usually playful and carefree.

- **Culture Connoisseur**
  - Your perfect destination offers an abundance of art, architecture, galleries, and theaters.

- **Beach Bum**
  - Your ideal trip revolves around enjoying the latest water sports, sipping tropical drinks, and working on your tan.

If you did not find a personality that fits you, build your own travel personality.
Welcome to TripMatcher™, the Web's first vacation advisor. We have researched all the major resorts for you. Answer a few simple questions, and we'll suggest the ski resort that best matches your preferences.

No time to answer? Click here . . .

<table>
<thead>
<tr>
<th>Activities</th>
<th>Optional Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>What do you enjoy?</td>
<td>You may refine your search.</td>
</tr>
<tr>
<td>☑ Adventure Sports</td>
<td>☑ Avoid Crowded Destinations</td>
</tr>
<tr>
<td>☑ Relaxing</td>
<td>☑ Avoid Jet Lag</td>
</tr>
<tr>
<td>☑ Dining Out</td>
<td>☑ Choose Weather Conditions</td>
</tr>
<tr>
<td>☑ Leisure Activities</td>
<td>☑ Good Safety Conditions</td>
</tr>
<tr>
<td>☑ Nightlife</td>
<td>☑ Improve A Foreign Language</td>
</tr>
<tr>
<td>☑ Shopping</td>
<td>☑ Select A Specific Environment</td>
</tr>
<tr>
<td>☑ Sights &amp; Culture</td>
<td>☑ Set A Budget</td>
</tr>
<tr>
<td>☑ Theme Parks &amp; Zoos</td>
<td>☑ Specify A Region</td>
</tr>
<tr>
<td>☑ Water Sports</td>
<td>☑ Traveler Support</td>
</tr>
<tr>
<td>☑ Winter Sports</td>
<td>☑ Traveling With Companions</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Timing</th>
<th>Departure City</th>
</tr>
</thead>
<tbody>
<tr>
<td>When are you leaving?</td>
<td>Please choose your gateway:</td>
</tr>
<tr>
<td>Late November</td>
<td>Washington Dc</td>
</tr>
<tr>
<td>How long will you be gone?</td>
<td></td>
</tr>
<tr>
<td>One Week</td>
<td></td>
</tr>
</tbody>
</table>
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?
  - Children's Adventure Sports
  - Hiking
  - Mountain Biking
  - Paragliding
  - Rock Climbing
  - Whitewater Rafting

**Relaxing**
- Which of these do you enjoy?
  - Enjoying Spa Treatments
  - Lying On A Beach
  - Sitting In Cafes
  - Strolling In Parks
  - Watching Sports

**Save Your Preferences**

Email Address:  

visitorx@yahoo.com
We found 15 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...more

★★★★★ Sitting In Cafes   ★★★★★ Hiking

(Flying time: 5 hours)

2. Salem And The North Shore - Massachusetts
Imagine for a moment that you could disintegrate yourself, Willy Wonka 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

★★★★★ Sitting In Cafes   ★★★★★ Hiking

(Flying time: 2 hours)

3. Marin County - California
Often dubbed the "Bay Area's 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...more

★★★★★ Sitting In Cafes   ★★★★★ Hiking

(Flying time: 7 hours)
Matching in TripleHop

Example: TripleHop

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

Catalogue of Destinations

[Delgado and Davidson, 2002]
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.
Learning User Profile: query mining

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

C-UM:00357
- activities
  - relaxing
  - sleeping
  - shopping
- constraint
  - meat = pork
  - budget = 200

Old query – user model

New user request, as computed by the systems. Shadowed means less important.
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” (*See Part 8 – Query expansion – based on co-occurrence analysis in the corpus of documents*)
  - 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)

[Aamodt and Plaza, 1994]
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”
Instance-based learning – Lazy Learning

- One way of solving tasks of approximating discrete or real valued target functions
- Have training examples: \((x_n, f(x_n)), n=1,...,N\)
- 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.
The distance between examples

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

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

<table>
<thead>
<tr>
<th>Number</th>
<th>Lines</th>
<th>Line types</th>
<th>Rectangles</th>
<th>Colours</th>
<th>Mondrian?</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6</td>
<td>1</td>
<td>10</td>
<td>4</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>2</td>
<td>8</td>
<td>5</td>
<td>No</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>2</td>
<td>7</td>
<td>4</td>
<td>Yes</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>1</td>
<td>8</td>
<td>4</td>
<td>Yes</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>1</td>
<td>10</td>
<td>5</td>
<td>No</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
<td>8</td>
<td>6</td>
<td>Yes</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>1</td>
<td>14</td>
<td>5</td>
<td>No</td>
</tr>
</tbody>
</table>

## Test instance

<table>
<thead>
<tr>
<th>Number</th>
<th>Lines</th>
<th>Line types</th>
<th>Rectangles</th>
<th>Colours</th>
<th>Mondrian?</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>7</td>
<td>2</td>
<td>9</td>
<td>4</td>
<td></td>
</tr>
</tbody>
</table>
Feature values are not normalized

Feature values are normalized

What is the difference between this feature values normalization and vector normalization in IR?

\[ x' = \frac{(x - \text{avg}(X))}{4 \times \text{stdev}(X)}, \text{ where } x \text{ is a feature value of the feature } X \]
Example of CBR Recommender System

- Entree is a restaurant recommender system – it finds restaurants:
  1. matching some user goals (case features)
  2. or similar to restaurants the user knows and likes
The Product is the Case

- In Entrée a case is a restaurant – the case is the product
- The problem component is the description of the restaurant given by the user
- The user will input a partial description of it – this is the only difficulty
- The solution part of the case is the restaurant itself – i.e. the name of the restaurant
- The assumption is that the needs of the user can be modeled as the features of the product description ....
Partial Match

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

The Washington DC restaurant you chose is:

Parioli
4800 Elm St. (Wisconsin Ave.), Bethesda, MD, 301-951-8600

Italian
$15-$30
Excellent Decor, Excellent Service, Extraordinary Food, Authentic,
Catering for Special Events, Takeout Available, Delivery Available,
Health Conscious Menus, Dining Outdoors, Parking/Valet, Private Rooms
Available, Private Parties, No Smoking Allowed, Weekend Dining,
Wheelchair Access

We recommend:

Stefani's (map)
1418 W. Fullerton Ave. (Southport Ave.), Chicago, 312-348-0111 601
Skokie Blvd. (bet. Dundee & Lake Cook Rds.), Northbrook,
708-564-3950

Pizza, Italian
$15-$30
Excellent Decor, Excellent Service, Excellent Food, Dining Outdoors,
Private Rooms Available, Private Parties, Weekend Brunch, Parking/Valet
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.
  - *Typically the relaxation of constraints will produce many restaurants in the result set.*
- Sorts the retrieved cases using a similarity metric – this takes into account all the input features.
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 a Utility-Similarity defined as the sum of local utilities.
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 `cuisine=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):
  - $\text{Sim}(q, \text{Dolce}) = 0.5 \times (1 - 1/30) + 0.3 \times 0 + 0.2 \times 0 = 0.48$
  - $\text{Sim}(q, \text{Gabbana}) = 0.5 \times (1 - 3/30) + 0.3 \times 1 + 0.2 \times 1 = 0.45 + 0.3 + 0.2 = 0.95$
NutKing

Travel Plan

Please tell us what you'd like to do on this trip. Your answers will help the system to make the best possible recommendations. (The answers you give apply only to this trip. Why?)

Tip: If you'd like to save your travel plans, please register now.

<table>
<thead>
<tr>
<th>TRAVEL COMPANIONS</th>
<th>DEPARTURE</th>
<th>PERIOD</th>
<th>ACTIVITIES</th>
</tr>
</thead>
<tbody>
<tr>
<td>Who will you travel with?</td>
<td>Where are you from?</td>
<td>When do you want to travel?</td>
<td>What would you like to do on this trip?</td>
</tr>
<tr>
<td>with family</td>
<td>Italy</td>
<td>August</td>
<td>□ Sports</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TRANSPORT</th>
<th>ACCOMMODATION</th>
<th>PREVIOUS VISITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>How will you travel?</td>
<td>What kind of accommodations do you want?</td>
<td>Have you ever visited Trentino?</td>
</tr>
<tr>
<td>car</td>
<td>hotel</td>
<td>a few times</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>between 20 and 40 €</th>
<th>NEXT</th>
</tr>
</thead>
</table>

http://www.visitfinland.com/web/guest/travel-planner/home
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 "Car park" to your query.
- Add "TV" 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:

- Trying to remove "Location" from the research and you obtain 15 results. Click on Remove and Get results to view.
- Trying to modify "Cost" from the research and you obtain 2 results. Click on Modify and Get results to view.
- Trying to remove "Outdoor swimming pool" from the research and you obtain 1 result. Click on Remove and Get results to view.
- Trying to remove "Solarium" from the research and you obtain 6 results. Click on Remove and Get results to view.
Suggested Travel Plan

Here's a trip we recommend. If you like you can save this as your trip.

Garda Lake in July

**General Information**

- **Name:** Garda Lake in July
- **Description:**
- **Start date:** 2002-07-01
- **End date:** 2002-07-13
- **Travel plan preferences**
  - **Travel companions:** with family
  - **Activities:** Sports, Reliving, Wine and Food, Environment and Landscape
  - **Accommodation:** apartment between 20 and 40 €
  - **Transport:** car
  - **Period:** July
  - **Length of stay:** two weeks

**The travel plan includes:**

Vacanza in montagna 2002

- **Description:** Con la famiglia nel Prinero
- **From:** 2002-08-05
- **To:** 2002-08-15
- **Locations:** MER
- **Accommodation:** AL BIVIO
- **Sporting activities:** Passeggiate Passo Respert
- **Culture:** Palazzo del Dazio o delle Miniere; Castel Pietra; Museo della Grande Guerra

TRAVEL 09.12.2002

- **Description:** val di Fassa
- **From:** 2002-07-01
- **To:** 2002-07-07
- **Accommodation:** RESIDENCE DOLOMIA
- **Sporting activities:** Pozze-Duffere-Sella
- **Culture:** Torre di Pozza; Moin de Pozzo

---

**Accommodation:** RESIDENCE SPIAGGIA

**Details**

- **Address:**
- **Telephone:**
- **Fax:**
- **Web:** www.rivadelgarda.com/spaggi
- **E-mail:**
- **Max Cost:** 25 €
NutKing as a CBR System

- **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
Travel Plan and Interaction Session Model

Collaborative Component 1: travel wish

- clf (family, bdg_medium, 7, Hotel)

Queries on content attributes

- cnq (Golfing=True AND Nightlife=True)
- cnq (category=3 AND Health=True)

Collaborative Component 2: selected products

- Travel bag (Kitzbühel, True, True, ..)
- rating (Hotel Schwarzer, 3, True, ..)

July in Fiemme Valley

Thanks for planning your trip with us. Remember that you can review details of your trip by clicking on My Travels. Bon voyage.
Item Ranking

1. Search the catalogue

2. Search Similar Cases

3. Output Reference Set

4. Sort locations $\text{loc}_i$ by similarity to locations in reference cases

Interactive query management

Suggest Q changes

Input

Output

Current Case

Travel components

Locations from Catalogue

Ranked Items
Two-fold Similarity

Product similarity

Target user

Target session
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 \((c_1, \ldots, c_{10})\) 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 \((p_1, \ldots, p_{10})\) from cases \((c_1, \ldots, c_{10})\) of the same type as those in \( R \)
  3. For each product \( r \) in \( R \) compute the \( \text{Score}(r) \) as the maximum 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 \( \text{Score}(r) \).
Example: Scoring Two Destinations

Destinations matching the user’s query

Score(D_i) = Max_j {Sim(CC,C_j)*Sim(D_i,CD_j)}

<table>
<thead>
<tr>
<th>Sim(CC,C1)</th>
<th>0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sim(CC,C2)</td>
<td>0.6</td>
</tr>
</tbody>
</table>

| Sim(D1, CD1) | 0.4 |
| Sim(D1, CD2) | 0.7 |
| Sim(D2, CD1) | 0.5 |
| Sim(D2, CD2) | 0.3 |

Score(D1)=Max{0.2*0.4, 0.6*0.7}=0.42
Score(D2)=Max{0.2*0.5, 0.6*0.3}=0.18
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
### Item Representation

<table>
<thead>
<tr>
<th></th>
<th>Node Type</th>
<th>Metric Type</th>
<th>Example: Canazei</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_1$</td>
<td>LOCATION</td>
<td>Set of hierarchical related symbols</td>
<td>Country=ITALY, Region=TRENTINO, TouristArea=FASSA, Village=CANAZEI</td>
</tr>
<tr>
<td>$X_2$</td>
<td>INTERESTS</td>
<td>Array of Booleans</td>
<td>Hiking=1, Trekking=1, Biking=1</td>
</tr>
<tr>
<td>$X_3$</td>
<td>ALTITUDE</td>
<td>Numeric</td>
<td>1400</td>
</tr>
<tr>
<td>$X_4$</td>
<td>LOCTYPE</td>
<td>Array of Booleans</td>
<td>Urban=0, Mountain=1, Rivereside=0</td>
</tr>
</tbody>
</table>

**TRAVELDESTINATION:** $(X_1, X_2, X_3, X_4)$

- $X_1 = (\text{Italy, Trentino, Fassa, Canazei})$
- $X_2 = (1, 1, 1)$
- $X_3 = 1400$
- $X_4 = (0, 1, 0)$

**dest1**
Item Query Language

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

\[ X_1 = (\text{Italy, Trentino, Fassa, Canazei}) \]
\[ X_2 = (1,1,1) \]
\[ X_3 = 1400 \]
\[ X_4 = (0, 1, 0) \]

- 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

\[
\begin{cases}
  x_{ik} = \text{true} & \text{if } x_{ik} \text{ is boolean} \\
  x_{ik} = v & \text{if } x_{ik} \text{ is nominal} \\
  l \leq x_{ik} \leq u & \text{if } x_{ik} \text{ is numerical}
\end{cases}
\]
Item Similarity

If $X$ and $Y$ are two items with same node-type:

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

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

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

$$\text{Sim}(X,Y) = 1 - d(X,Y) \text{ or } \text{Sim}(X,Y) = \exp(-d(X,Y))$$
**Item Similarity Example**

\[
\begin{align*}
X_1 &= (I, \text{TN}, \text{Fassa}, \text{Canazei}) \\
X_2 &= (1,1,1) \\
X_3 &= 1400 \\
X_4 &= (0, 1, 0) \\
Y_1 &= (I, \text{TN}, \text{Fassa},?) \\
Y_2 &= (1,0,1) \\
Y_3 &= 1200 \\
Y_4 &= (1, 1, 0)
\end{align*}
\]

**Sim(dest_1, dest_2)**

\[
\begin{align*}
&= \exp\left(-\frac{1}{\sqrt{4}} \sqrt{d_1(X_1, Y_1)^2 + \cdots + d_4(X_4, Y_4)^2}\right) \\
&= \exp\left(-\frac{1}{\sqrt{4}} \sqrt{(0.3)^2 + (1 - 2/3)^2 + ((1400 - 1200)/2000)^2 + (1 - 1/2)^2}\right) \\
&= \exp\left(-\frac{1}{\sqrt{4}} \sqrt{0.461}\right) = \exp(-0.339) = 0.712
\end{align*}
\]

3 in the union 2 in the union
Case Distance

- nt: case
  - mt: vector
    - c1
      - cart1
        - clf1
          - cnq1
      - clf1
        - cnq1

- nt: destinations
  - mt: set
    - dests1
      - accs1
        - acts1
    - dests2
      - accs2
        - acts2

- nt: destination
  - mt: vector
    - dest1
      - X1
      - X2
      - X3
      - X4
    - dest2
      - Y1
      - Y2
      - Y3
      - Y4

- nt: location
  - mt: hierarchical
Case Distance

\[ d(c_1, c_2) = \frac{1}{\sqrt{\sum_{i=1}^{3} W_i}} \sqrt{W_1 d(cart_1, cart_2)^2 + W_2 d(clf_1, clf_2)^2 + W_3 d(cnq_1, cnq_2)^2} \]
\[ d(cart_1, cart_2) = \sqrt{d(dests_1, dests_2)^2 + d(accs_1, accs_2)^2 + d(act_1, act_2)^2} \]
\[ d(dests_1, dests_2) = \frac{1}{2 \times 3} (d(dest_1, dest_3) + d(dest_1, dest_4) + d(dest_1, dest_5) + d(dest_2, dest_3) + d(dest_2, dest_4) + d(dest_2, dest_5)) \]
CBR Knowledge Containers

- CBR is a knowledge-based approach to problem solving
- The knowledge is “contained” into four containers
  - **Cases**: the instances belonging to our case base
  - **Case representation language**: the representation language that we decided to use to represent cases
  - **Retrieval knowledge**: the knowledge encoded in the similarity metric and in the retrieval algorithm
  - **Adaptation knowledge**: how to reuse a retrieved solution to solve the current problem.
Conclusions

- Knowledge-based systems exploits knowledge to map a user to the products she likes
- KB systems uses a variety of techniques
- Knowledge-based systems requires a big effort in term of knowledge extraction, representation and system design
- Many KB recommender systems are rooted in Case-Based Reasoning
- Similarity of complex data objects is required often required in KB RSs.
- NutKing is a hybrid case-based recommender system
- The case is the recommendation session.
Questions

- What are the main differences between a CF recommender system and a KB RS (such as activebuyers.com or Entree)?
- What is the role of query augmentation?
- What is the basic rationale of a CBR recommender system?
- What is a case in a CBR recommender system such as Entree?
- How a CBR recommender system learns to recommend?
- What are the knowledge containers is a CBR RS?
- What are the main differences between a “classical” CBR recommender system such as Entrée and Nutking?
- What are the motivations for the introduction of the double-similarity ranking method?
- What are the types of local similarity metrics used in Nutking?