Content-Based Filtering
and
Hybrid Methods

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

Content

- Typologies of recommender systems
- Content-based recommenders
- Naïve Bayes classifiers and Content-based filtering
- Content representation (bag of words, tf-idf)
- Demographic-based recommendations
- Clustering Methods
- Utility-based Methods
- Hybrid methods

Other Recommendation Techniques

- The distinction is not related to the user interface or the properties of the user's interaction but rather the source of data used for the recommendation
- **Background data:** the information of the system before the recommendation process starts
- **Input data:** the information that the user must communicate to the system to get a recommendation
- **The algorithm:** that combines background and input data to build a recommendation

[Burke, 2002]

3

"Core" Recommendation Techniques

U is a set of users
I is a set of items/products

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**, and 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.

[Burke, 2002]

4

Content-Based Recommendation

- In *content-based* recommendations the system tries to recommend items similar to those a given user has liked in the past
- In contrast with *collaborative* recommendation where the system identifies users whose tastes are similar to those of the given user and recommends items *they* have liked
- A **pure** content-based recommender system makes recommendations for a user based solely on the profile built up by analyzing the content of items **which that user has rated in the past.**

Simple Example

- I saw yesterday “Harry Potter and and the Sorcerer's Stone”

- The recommender system suggests:
  - Harry Potter and the Chamber of Secrets
  - Polar Express

Content-Based Recommender

- Has its root in **Information Retrieval** (IR)
- It is mainly used for recommending **text-based products** (web pages, usenet news messages,)
- The items to recommend are “described” by their associated **features** (e.g. keywords)
- The **User Model** can be structured in a “similar” way as the content: for instance the features/keywords more likely to occur in the preferred documents (lazy approach)
  - Then, for instance, text documents can be recommended based on a comparison between their content (words appearing in the text) and a user model (a set of preferred words)
- The user model can also be a **classifier** based on whatever technique (Neural Networks, Naïve Bayes, C4.5,)

7

Stable and Ephemeral Preferences

- The user model **typically** describes stable preferences – since it is build by mining (all) previous user-system interaction (ratings or queries)
- But one can build a content-based recommender system, more similar to an IR system, acquiring on-line the user model (query)
- Or stable preferences and short-term ones can be combined.

8

Syskill & Webert [Pazzani &Billsus, 1997]

- Assisting a person to find information that satisfies long-term, recurring goals (e.g. digital photography)
- Feedbacks on the “interestingness” of a set of previously visited sites is used to learn a profile
- The profile is used to predict interestingness of unseen sites.

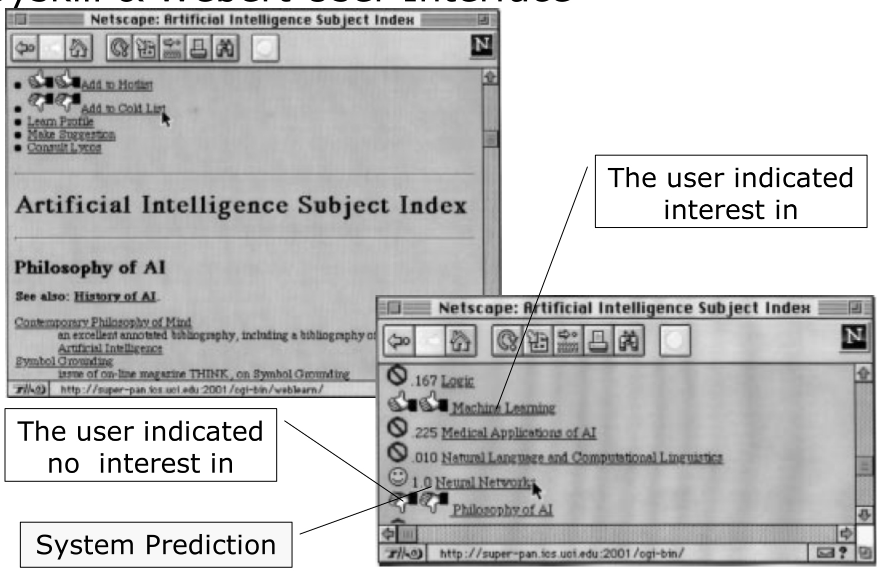
9

Supported Interaction

- The user identifies a topic (e.g. Biomedical) and a page with many links to other pages on the selected topic (index page)
- The user can then explore the Web with a browser that in addition to showing a page
  - Offers a tool for collecting user ratings on displayed pages
  - Suggests which links on the current page are (estimated) interesting.

10

Syskill & Webert User Interface



11

Content Model

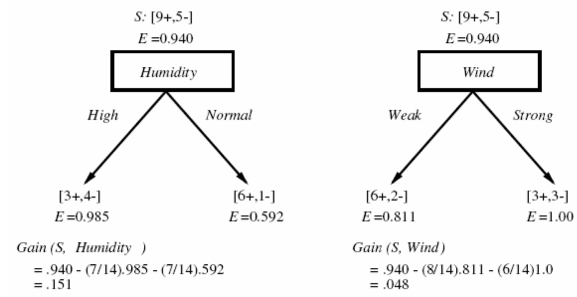
- A document (HTML page) is described as a set of Boolean features (a word is present or not)
- A feature is considered important for the prediction task **if the Information Gain is high**
- G(S,W) = E(S) –[P((W is present)E($S_{\text{W is present}}$) + P(W is absent)E($S_{\text{W is absent}}$)]

$$E(S) = \sum_{c \in \{hot, cold\}} - p(S_c) \log_2(p(S_c))$$

- E(S) is the Entropy of a labeled collection (how randomly the two labels are distributed)
- W is a word – a Boolean feature (present/not-present)
- S is a set of documents, S_{hot} is the subset of interesting documents
- They have used the 128 most informative words (highest information gain).

12

Entropy and Information Gain example



Smaller Entropy
Higher Information Gain

- 9 positive and 5 negative examples → E(S)=0.940
- Using the "humidity" attribute – the entropy of the split produced is:
  - P((Humidity is high)E($S_{\text{hum. is high}}$) + P(Humidity is normal)E($S_{\text{hum. is normal}}$)=(7/14)*0.985 + (7/14)*0.592 = 0.789
- Using the "wind" attribute – the entropy of the split produced is:
  - P((wind is weak)E($S_{\text{wind. is weak}}$) + P(wind is strong)E($S_{\text{wind is strong}}$)=(8/14)*0.811 + (6/14)*1.0 = 0.892

13

Learning

- They used a Bayesian classifier (one for each user), where the probability that a document $w_1=v_1$, …, $w_n=v_n$ (e.g. car=1, story=0, …, price=1) belongs to a class (cold or hot) is

$$P(C = hot \mid w_1 = v_1, \ldots, w_n = v_n) \cong P(C = hot) \prod_j P(w_j = v_j \mid C = hot)$$

- Both $P(w_j = v_j|C{=}hot)$ (i.e., the probability that in the set of the documents liked by a user the word w_j is present or not) and P(C=hot) is estimated from the training data
- After training on 30/40 examples it can predict hot/cold with an accuracy between 70 and 80 %

14

A Better Model for the Document

- TF-IDF means Term Frequency – Inverse Document Frequency

$$d_{t_i} = \left(0.5 + 0.5\frac{tf_i}{tf_{max}}\right)\left(\log\frac{n}{df_i}\right)$$

The less frequent the word is in the corpus the greater is this

The greater the frequency of the word the greater is this term

- tf_i is the number of times word t_i appears in document d (the term frequency),
- df_i is the number of documents in the corpus which contain t_i (the document frequency),
- n is the number of documents in the corpus and tf_{max} is the maximum term frequency over all words in d.

15

Using TF-IDF

- One can build a classifier (e.g. Bayesian) as before, where instead of using a Boolean array for representing a document, the array now contains the tf-idf values of the selected words (a bit more complex because features are not Boolean anymore)
- But can also build a **User Model** by (Rocchio, 1971)
  - Average of the tf-idf representations of interesting documents of a user (Centroid)
  - Subtracting a fraction of the average of the not interesting documents (0.25 in [Pazzani & Billsus, 1997]
- **Then new documents close (cosine distance) to this user model are recommended.**

16

Problems of Content-Based Recommenders

- A very shallow analysis of certain kinds of content can be supplied
- Some kind of items are not amenable to any feature extraction methods with current technologies (e.g. movies, music)
- Even for texts (as web pages) the IR techniques cannot consider multimedia information, aesthetic qualities, download time
  - Hence if you rate positively a page it could be not related to the presence of certain keywords!

17

Problems of Content-Based Recommenders (2)

- **Over-specialization:** the system can only recommend items scoring high against a user's profile – the user is recommended with items similar to those already rated
- **Requires user feed-backs:** the pure content-based approach (similarly to CF) **requires user feedback** on items in order to provide meaningful recommendations
- **It tends to recommend expected items** – this tends to **increase trust** but could make the recommendation not much useful (no serendipity)
- Works better in those situations where the **"products" are generated dynamically** (news, email, events, etc.) and there is the need to check if these items are relevant or not.

18

"Core" Recommendation Techniques

U is a set of users
I is a set of items/products

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**, and 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.

[Burke, 2002]

19

Demographic Methods

- Aim to categorize the user based on personal attributes and make recommendation based on demographic classes
- Demographic groups can come from marketing research – hence experts decided how to model the users
- Demographic techniques form people-to-people correlations
- Tend to be similar to collaboration via content (we shall discuss it later) but does not use explicit ratings

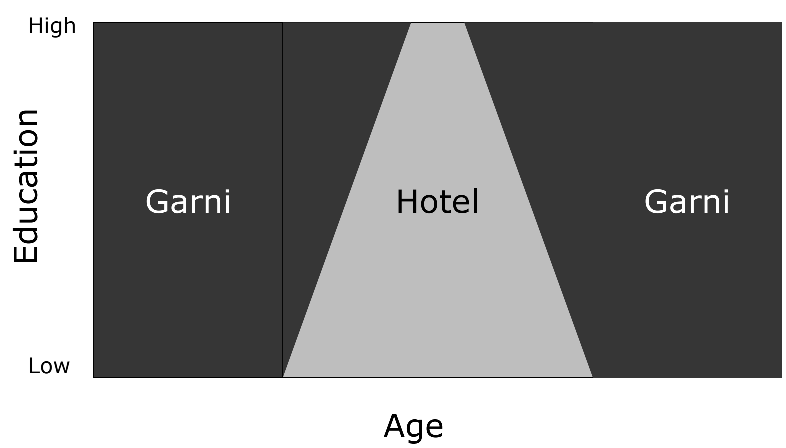
20

Simple Demographic Method

- The marketer knows how to separate the demographic classes and exploits this knowledge to define the **personalization rules**
- This is the method used by many commercial (expensive) personalization engines (e.g. ATG) [Fink & Kobsa, 2000]
- It is very efficient but:
  - Do not tracks the changes in the population (user products)
  - Rely on the rules inserted by an "expert"
  - Suffers of all the classical problems of Expert Systems (e.g. brittle)

21

Example



22

Demographic-based personalization

23

Demographic-based personalization

24

Demographic Methods (more sophisticated)

	gender	age	area code	education	employed	Dolce
Karen	F	15	714	HS	F	+
Lynn	F	17	714	HS	F	–
Chris	M	35	714	C	T	+
Mike	F	40	714	C	T	–
Jill	F	10	714	E	F	?

- Demographic features in general are asked
- But can also induced by classifying a user using other user descriptions (e.g. the home page) – you need some user for which you know the class (e.g. male/female)
- Prediction can use whatever learning mechanism we like (nearest neighbor, naïve classifier, etc.)

[Pazzani, 1999]

25

Clustering Methods

- Use a clustering method to divide the customer base into segments
  - Unsupervised method
  - Using a similarity measure for customers
  - Typically a greedy algorithm
- Assign each user to a cluster – the one that contains the most similar user
- Use purchases or ratings of customers in the segment to generate recommendations
- Many different user models can be used for the similarity computation

26

"Core" Recommendation Techniques

U is a set of users
I is a set of items/products

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**, and 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.

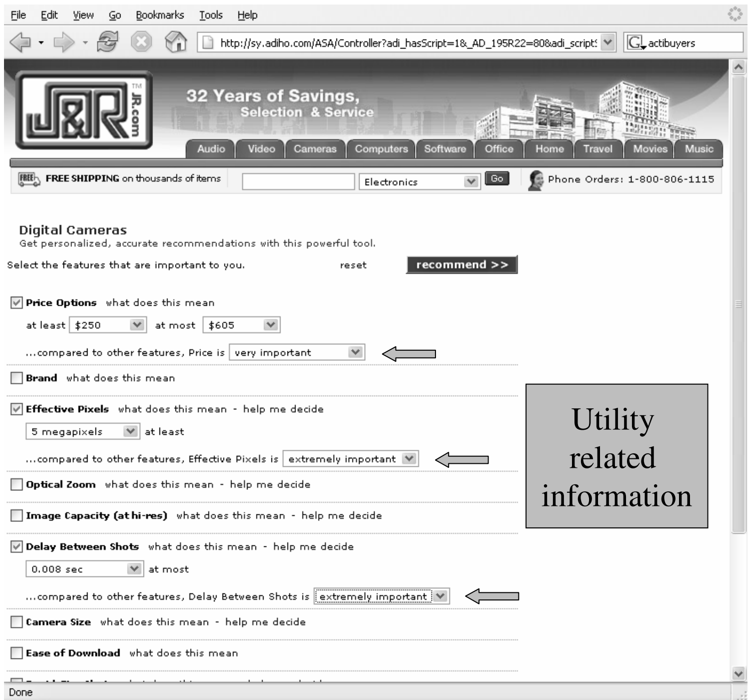
[Burke, 2002]

27

Utility methods

- A utility function is a map from a state onto a real number, which describes the associated degree of happiness
- Can build a long term utility function but more often the systems using such an approach try to acquire a short term utility function
- They **must acquire** the user utility function, or the parameters defining such a function

28



29

Utility

- The item is described by a list of attributes (numerical) p_1, …p_m, e.g., number of rooms, square meters, levels, (MaxCost – Cost), …
- It is generally assumed that higher values of the attribute correspond to higher utilities
- The user is modeled with a set of weights, u_1, …, u_m (in [0,1]) on the same attributes

$$U(u_1,\ldots,u_m,p_1,\ldots,p_m)=\sum_{j=1}^{m}u_j p_j$$

- The objective is to find (retrieve) the products with larger utility (maximal)
- The problem is the elicitation or learning of user model u_1, …, u_m

30

Utility and similarity

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

$$U(u_1, \ldots, u_m, q_1, \ldots, q_m, p_1, \ldots, p_m) = \sum_{j=1}^{m} u_j Sim(q_j, p_j)$$

- A typical local similarity function is (1-|q_j – p_j|/$range_i$), where $range_i$ is the difference between the max and min value of the attribute i.
- A utility-based recommender become 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).

31

Hybrid Methods

- Try to address the shortcomings of both approaches (content-based and collaborative-based) and produce recommendations using a combination of those techniques
- There is a large variability on these hybrid methods – there is no standard hybrid method
- We shall present some of them here but many will come later.

32

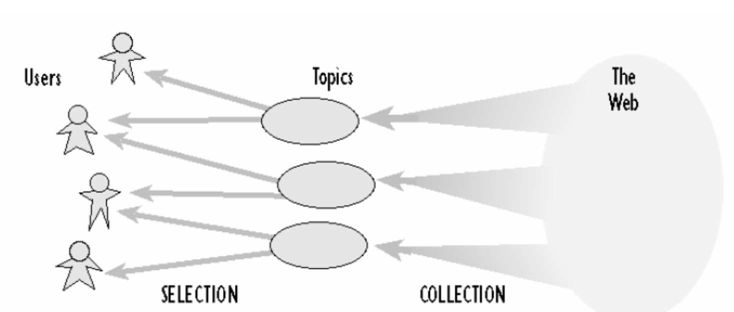
Fab System

- The user profile is based on content analysis
- Two user profiles are compared to determine similar users
- Then a collaborative based recommendation is generated
- User receive items **both** when they score highly against their own profile and when rated highly by a user with a similar profile
- In [Burke, 2002] terminology this a “mixed” approach (we’ll see it later)

[Balabanovich and Shoam, 1997]

33

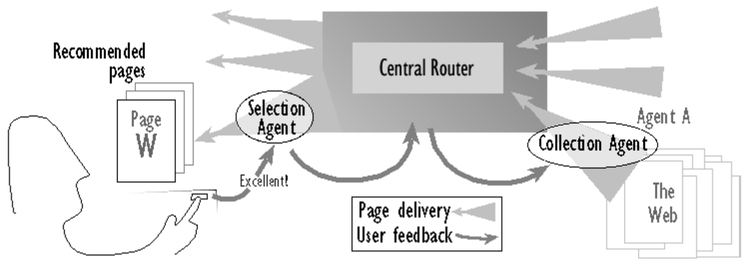
Selection of Relevant Pages



1. Pages relevant to specific topics are collected from the Web
2. Selections (recommendations) for individual users are made among these pages.

34

Fab Architecture



- Collection agents: find pages relevant for a specific topic
- Selection agent: find pages for a specific user
- Central router: forwards pages on to those users whose profiles they match above some threshold.

35

User Feedback

- When the user has requested, received, and looked over the recommendations, they are required to assign a rating (7-point scale)
- Ratings are store in the personal agent profile
- Ratings are sent to the collection agents for adapting the user profiles (stored by him)
- Highly rated pages are sent to similar users (collaborative)

36

Collection Agents

- The construction of accurate profiles is a key task (both for content-based and collaborative-based recommenders)
- The collection agents' profiles represent a topic of interest
- The population of collection agents adapts to the population of users
  - Unpopular collection agents (whose pages are not seen) or unsuccessful ones (whose pages receive low feedbacks) are removed
- This copes with the issue of keeping the profiles up-to-date

37

Empirical Evaluation (FAB)

- 11 users – each one interested in one topic
- The evaluate (feedback) on documents recommended by the system
- Every five sessions the users where shown a special selection used for evaluation purpose
- The ranking of the user evaluation was compared with the ranking conjectured by the system
- The two ranked list where compared with the **ndpm** measure

38

Yao's definition of: ndpm

- If $>_p$ is the ranking predicted by the recommender system, and $>_d$ is the user's desired ranking,
- Distance between two documents d d' :

$$\delta_{\succ_p,\succ_d}(\mathbf{d},\mathbf{d}') = \begin{cases} 2 & \text{if } (\mathbf{d} \succ_p \mathbf{d}' \text{ and } \mathbf{d}' \succ_d \mathbf{d}) \text{ or } (\mathbf{d} \succ_d \mathbf{d}' \text{ and } \mathbf{d}' \succ_p \mathbf{d}) \\ 1 & \text{if } (\mathbf{d} \succ_d \mathbf{d}' \text{ or } \mathbf{d}' \succ_d \mathbf{d}) \text{ and } \mathbf{d} \sim_p \mathbf{d}' \\ 0 & \text{otherwise} \end{cases}$$

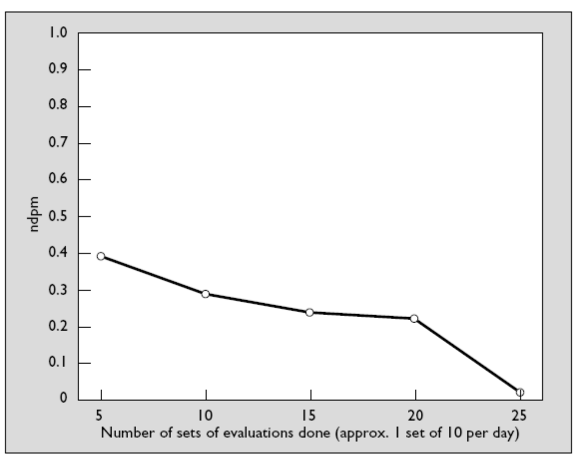
- Distance between two rankings ($|>_d|$ is the number of paired comparisons):

$$ndpm(\succ_p, \succ_d) = \frac{\sum_{\mathbf{d},\mathbf{d}' \in D} \delta_{\succ_p,\succ_d}(\mathbf{d},\mathbf{d}')}{2|\succ_d|}$$

39

[Balabanovich, 1998]

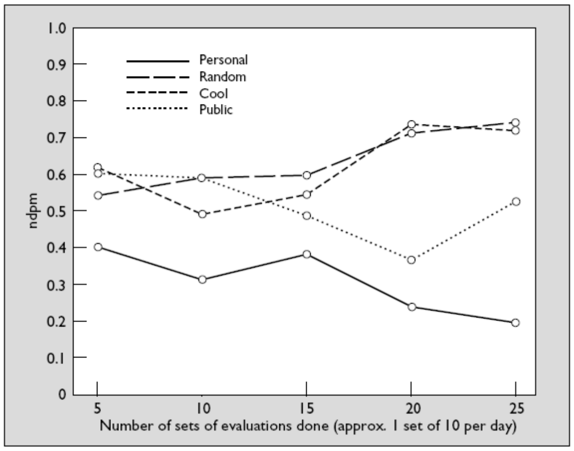
Ranking accuracy



- Distance between actual and predicted rankins – averaged over all users

40

Comparison with other approaches



- Personal = FAB personalised recommendations
- Random recommendations
- Cool = pages recommended by human (cool sites of the day)
- Public = best matching an average user profile (i.e. , is a non personalized solution)

41

Collaboration via Content

- **Problem addressed:** in a collaborative-based recommender, products rated by pair of users may be very few – correlation between two users is not reliable
- In collaboration via content the content-based profile of each user is exploited to detect similarities among users.

[Pazzani, 1999]

42

Content-Based Profiles

	noodle	shrimp	basil	exotic	salmon	Dolce
Karen	2.5	0	.2	0	0	+
Lynn	1.1	0	1.1	1.5	0	–
Chris	1.5	0	3.5	1.5	.5	+
Mike	1.1	1.1	2.1	2.0	2.5	–
Jill	1.1	2.2	0	0	3.5	?

- The weights can be the average of the TF-IDF vectors of the documents that are highly rated (as in FAB)
  - E.g. in the restaurants liked by Karen the word "noodle" is very frequent (and not much frequent in the entire collection of restaurant descriptions)
- Or you can use winnow as in [Pazzani, 1999]

43

Winnow

- Each word considered in the profiles is considered as a Boolean feature (present/not present)
- Winnowing learns a weight w_i associated to each word x_i
- Initially all the weights are set to 1. Then, for each document rated by the user a linear threshold function is computed

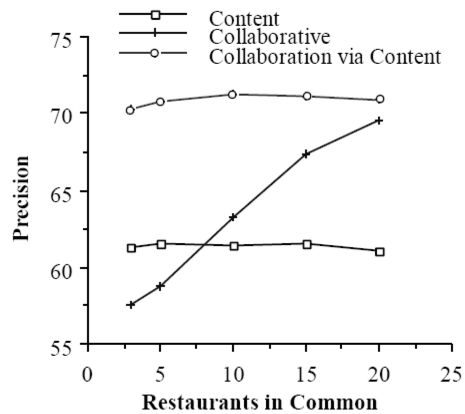
$$\sum w_i x_i > \tau$$

- If the above sum is over the threshold and the user did not liked then the weights associated with each word in the document are divided by 2
- If the sum is below the threshold and the user liked the document then all weights associated with words in the document are multiplied by 2
- Otherwise no change is made

[Pazzani, 1999]

44

Comparison



Averaged on 44 users

Precision is computed in the top 3 recommendations = (# of plus in the recommendation list)/3

- Content–based recommendation is done with Bayes classifier
- Collaborative is standard using Pearson correlation
- Collaboration via content uses the content-based user profiles built by winnow.

45

Details

- The target user training data are 30 restaurants, but they varied (from 3 to 20) those overlapping the other users in the training
- The content-based user model (used in the collaboration via content) is built with these 30 restaurant using winnow
- The Bayes classifier (content-based prediction) is built using the ratings of a user on these 30 restaurants (the ratings of the other users are never exploited)
- The collaborative recommender uses the rating of common restaurants (hence from 3 to 20 – and this is the only one affected by this parameter).

46

Hybrid methods

- More in general a hybrid method can be defined as a way to integrate two or more scores produced by two independent recommendation algorithms
- $S_A(p)$ is the predicted rating for product p computed by algorithm A
- $S_B(p)$ is the predicted rating for product p computed by algorithm B
- $S_H(p) = aS_A(p) + (1-a)S_B(p)$ hybrid rating (weighted)

47

Hybridization Methods

Hybridization method	Description
Weighted	The scores (or votes) of several recommendation techniques are combined together to produce a single recommendation.
Switching	The system switches between recommendation techniques depending on the current situation.
Mixed	Recommendations from several different recommenders are presented at the same time
Feature combination	Features from different recommendation data sources are thrown together into a single recommendation algorithm.
Cascade	One recommender refines the recommendations given by another.
Feature augmentation	Output from one technique is used as an input feature to another.
Meta-level	The model learned by one recommender is used as input to another.

[Burke, 2002]

48

Weighted

- The score of a recommended item is computed from the results of all of the available recommendation techniques present in the system
  - Example 1: a linear combination of recommendation scores
  - Example 2: treats the output of each recommender (collaborative, content-based and demographic) as a set of votes, which are then combined in a consensus scheme
- The implicit assumption in this technique is that the relative value of the different techniques is more or less uniform across the space of possible items
- Not true in general: e.g. a collaborative recommender will be weaker for those items with a small number of raters.

49

Weighted Example

- Movie recommendations that integrates item-to-item collaborative filtering and information retrieval [Park et al., 2006]
- **Information retrieval component:** Web(i, q)= (N+1-k(i))/N where N are the items returned by the query q and k(i) is the position of movie i in the results set (example q = “arnold swarzenegger”)
  - Movies highly ranked by the IR component (low k(i)) have a Web(i,q) value close to 1
- **Item-to-item collaborative filtering:** Auth(i, u) is the score of item i for user u
  - Movies similar to those higly ranked by the user in the past get a high Auth(i, u) score
- **Final rank:** MADRank(i,q,u)= a Auth(i,u) +(1-a)Web(i,q)
- If Auth(i,u) cannot be computed (not enough ratings for u or i) then Auth(I,u) can be a non personalized score (e.g. item popularity) or simply not used (also some switching!)

50

Switching

- The system uses some criterion to switch between recommendation techniques
- Example: The DailyLearner [Billsus and Pazzani, 2000] system uses a content/collaborative hybrid in which a content-based recommendation method is employed first
- If the content-based system cannot make a recommendation with sufficient confidence, then a collaborative recommendation is attempted
- This switching hybrid does not completely avoid the ramp-up problem, since both the collaborative and the content-based systems have the “new user” problem
- The main problem of this technique is to identify a GOOD switching condition.

51

Mixed

- Recommendations from more than one technique are presented together
- The mixed hybrid avoids the “new item” start-up problem
- It does not get around the “new user” start-up problem, since both the content and collaborative methods need some data about user preferences to start up.

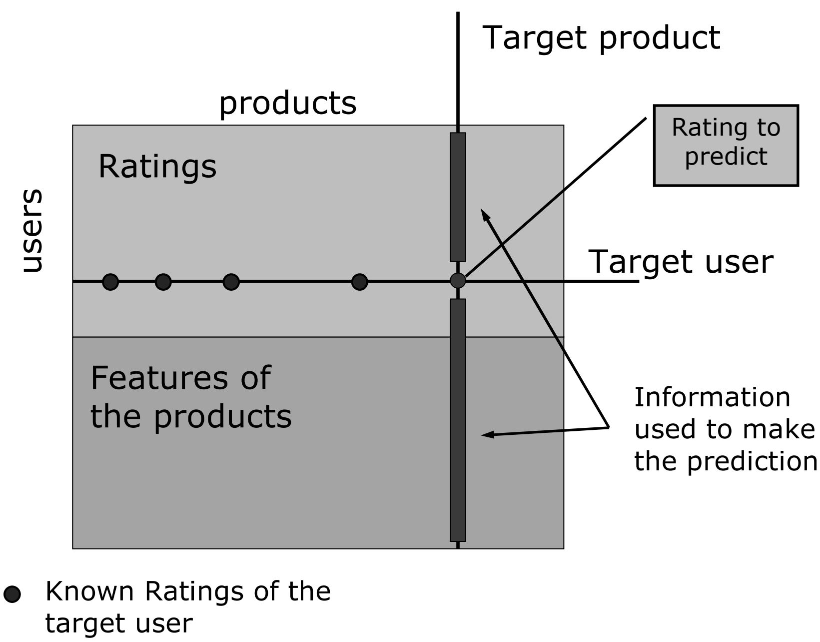
52

Feature Combination

- Achieves the content/collaborative merger treating collaborative information (ratings of users) as simply additional feature data associated with each example and use content-based techniques over this augmented data set
- [Basu, Hirsh & Cohen 1998] apply the inductive rule learner Ripper to the task of recommending movies using both user ratings and content features
- The feature combination hybrid lets the system consider collaborative data without relying on it exclusively, so it reduces the sensitivity of the system to the number of users who have rated an item
- The system have information about the inherent similarity of items that are otherwise opaque to a collaborative system.

53

Feature Combination



54

Cascade

- One recommendation technique is employed first to produce a coarse ranking of candidates and a second technique refines the recommendation from among the candidate set
- Example: EntreeC uses its knowledge of restaurants to make recommendations based on the user's stated interests. The recommendations are placed in buckets of equal preference, and the collaborative technique is employed to break ties
- Cascading allows the system to avoid employing the second, lower-priority, technique on items that are already well-differentiated by the first
- But requires a meaningful and constant ordering of the techniques.

55

Feature Augmentation

- Produce a rating or classification of an item and that information is then incorporated into the processing of the next recommendation technique
- Example: Libra system [Mooney & Roy 1999] makes content-based recommendations of books based on data found in Amazon.com, using a naive Bayes text classifier
- In the text data used by the system is included "related authors" and "related titles" information that Amazon generates using its internal collaborative systems
- Very similar to the feature combination method:
  - **Here** the output of a recommender system is used for a second RS
  - In **feature combination** the representations used by two systems are combined.

56

Meta-level

- Using the model generated by one as the input for another
- Example: FAB
  - user-specific selection agents perform content-based filtering using Rocchio's method to maintain a term vector model that describes the user's area of interest
  - Collection agents, which gather new pages from the web, use the models from all users in their gathering operations.
  - Documents are first collected on the basis of their interest to the community as a whole and then distributed to particular users
- Example: [Pazzani 1999] collaboration via content: the model generated by the content-based approach (winnow) is used for representing the users in a collaborative filtering approach.

57

Summary

- Content-based methods are well rooted in information retrieval
- Demographic methods are very simple and could provide limited personalization (sometime can be sufficient)
- Utility-based methods go to the root of the decision problem – how to acquire the utility function?
- Hybrid methods are the most powerful and popular right now – there are plenty of options for hybridization

58

Questions

- Can a content-based recommender operate in a not networked environment?
- List a set of attributes of a recommender system and compare a content-based system to a collaborative-based one.
- Is the Centroid of the interesting documents a good User Model? What are the problems of this representation? How to exploit ephemeral needs?
- How to build a content-based recommender for music or photography?
- Can a utility function learned or acquired without explicitly asking?
- Could you imagine different ways (not the sum) to integrate the utility over a single issue to produce the total utility?
- What are the pros and cons of different hybridization approaches?

59