Collaborative Filtering & Content-Based Recommending

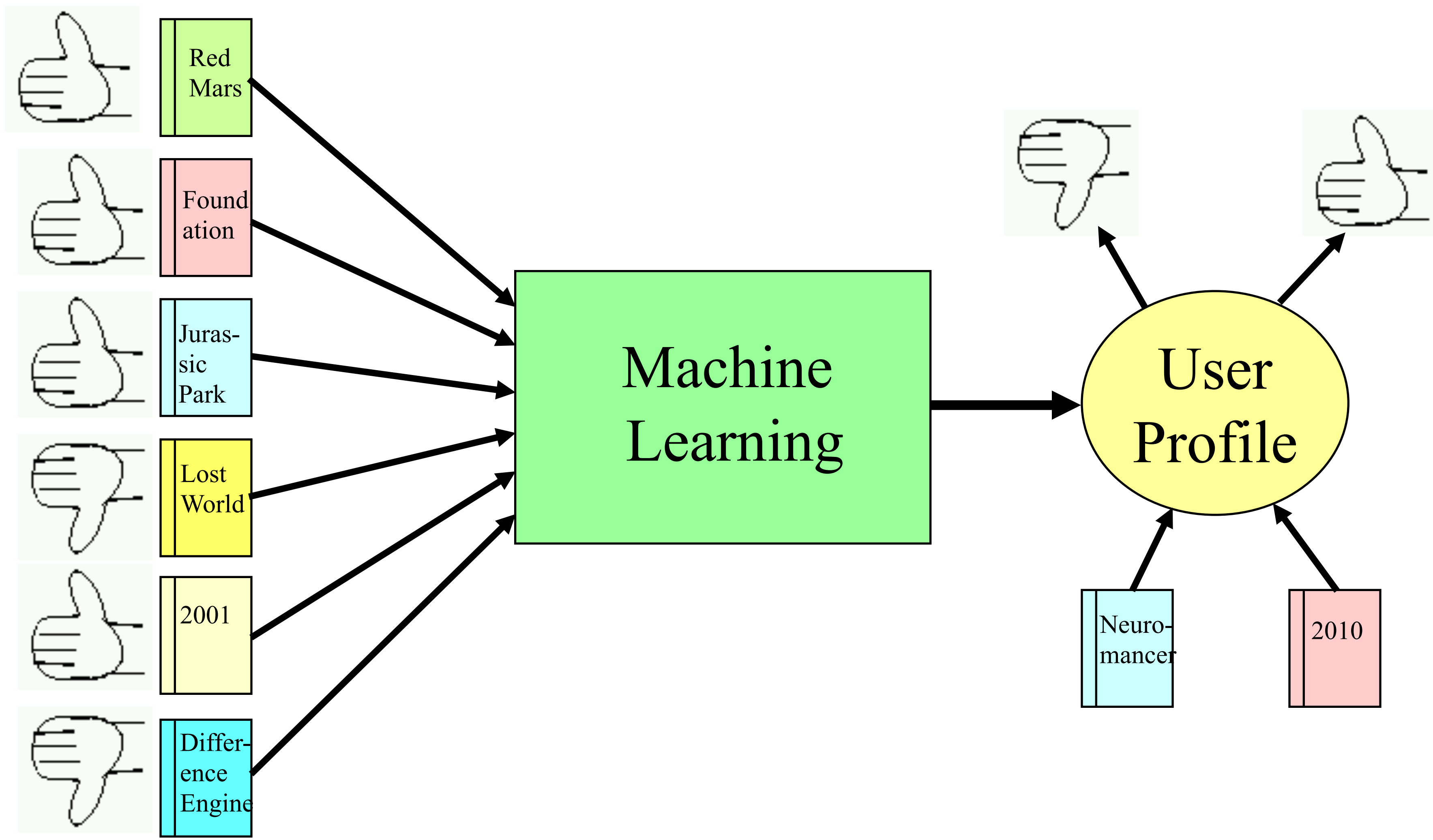
CS 293S. T. Yang

Slides based on R. Mooney at UT Austin

Recommendation Systems

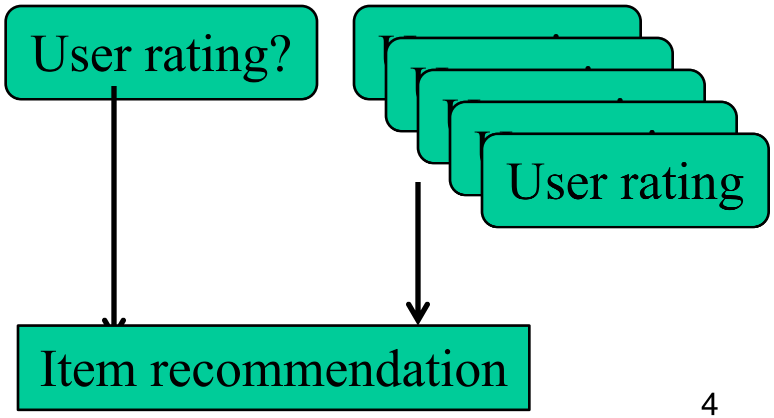
- Systems for recommending items (e.g. books, movies, music, web pages, newsgroup messages) to users based on examples of their preferences.
  - Amazon, Netflix. Increase sales at on-line stores.
- Basic approaches to recommending:
  - Collaborative Filtering (a.k.a. social filtering)
  - Content-based
- Instances of personalization software.
  - adapting to the individual needs, interests, and preferences of each user with recommending, filtering, & predicting

Process of Book Recommendation

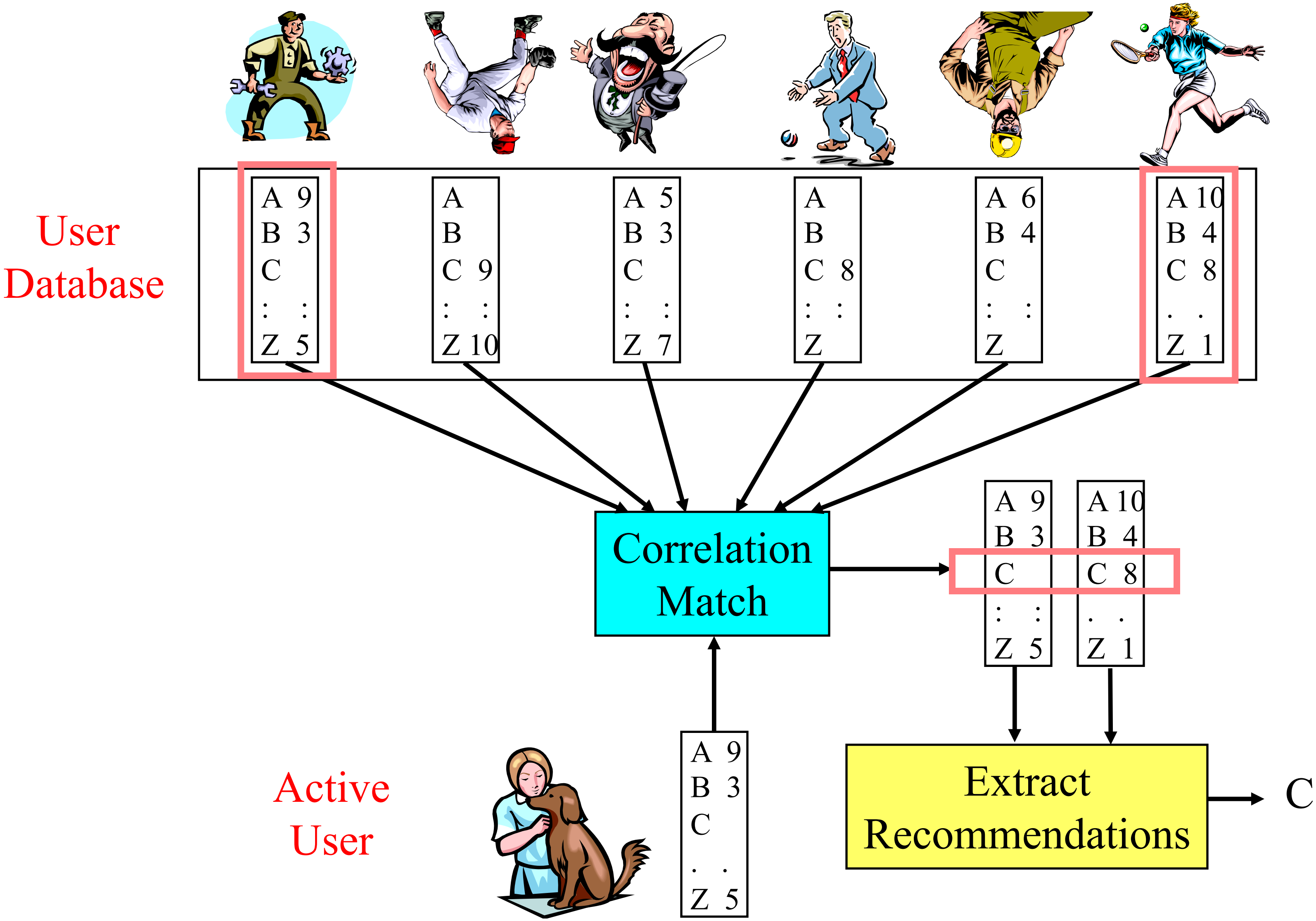


Collaborative Filtering

- Maintain a database of many users' ratings of a variety of items.
- For a given user, find other similar users whose ratings strongly correlate with the current user.
- Recommend items rated highly by these similar users, but not rated by the current user.
- Almost all existing commercial recommenders use this approach (e.g. Amazon).



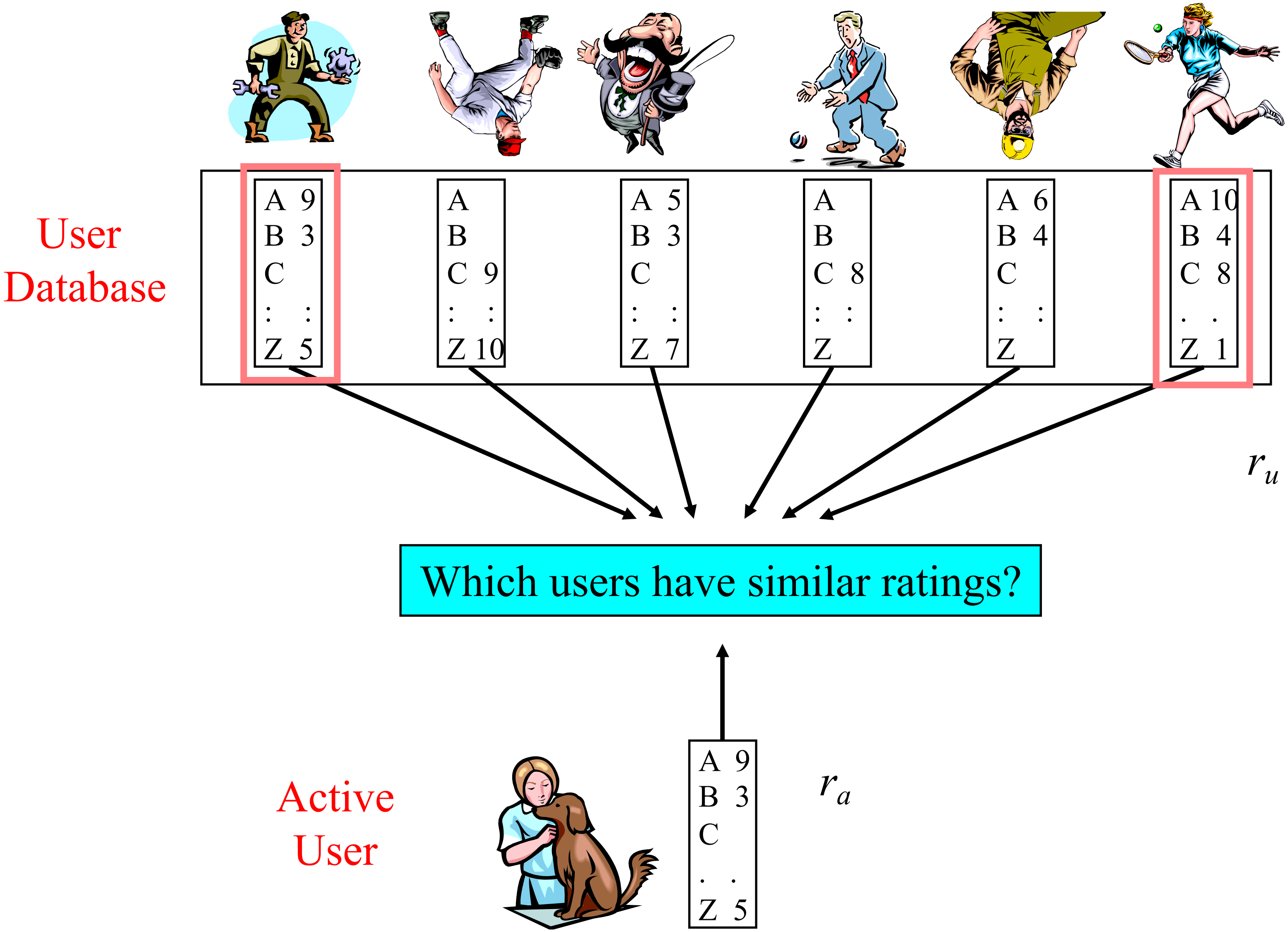
Collaborative Filtering



Collaborative Filtering Method

1. Weight all users with respect to similarity with the active user.
2. Select a subset of the users (*neighbors*) to use as predictors.
3. Normalize ratings and compute a prediction from a weighted combination of the selected neighbors' ratings.
4. Present items with highest predicted ratings as recommendations.

Find users with similar ratings/interests



Similarity Weighting

- Similarity of two rating vectors for active user, a, and another user, u.
  – Pearson correlation coefficient
  – a cosine similarity formula

$$c_{a,u} = \frac{\text{covar}(r_a, r_u)}{\sigma_{r_a}\sigma_{r_u}}$$

r_a and r_u are the ratings vectors for the m items rated by **both** a and u

User Database

A 9	A	A 5	A	A 6	A 10
B 3	B	B 3	B	B 4	B 4
C	C 9	C	C 8	C	C 8
: :	: :	: :	: :	: :	. .
Z 5	Z 10	Z 7	Z	Z	Z 1

Definition: Covariance and Standard Deviation

- Covariance:

$$\text{covar}(r_a, r_u) = \frac{\sum_{i=1}^{m} (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{m}$$

$$\bar{r}_x = \frac{\sum_{i=1}^{m} r_{x,i}}{m}$$

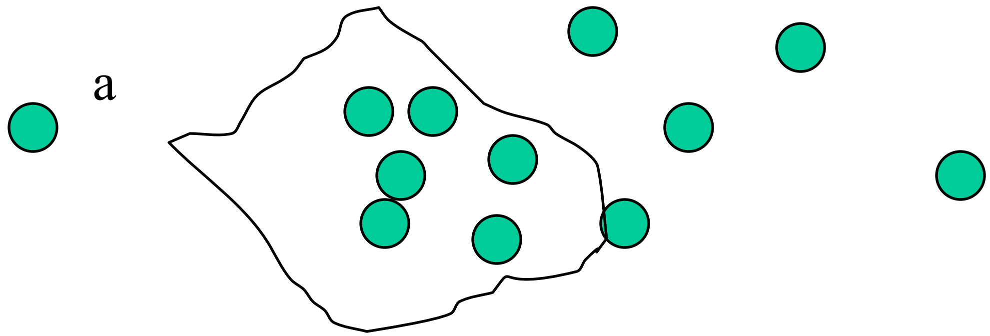
$$\sigma_{r_x} = \sqrt{\frac{\sum_{i=1}^{m} (r_{x,i} - \bar{r}_x)^2}{m}}$$

- Standard Deviation:
- Pearson correlation coefficient

$$c_{a,u} = \frac{\text{covar}(r_a, r_u)}{\sigma_{r_a} \sigma_{r_u}} = \text{Cosine}(r_a - \bar{r}_a, r_u - \bar{r}_u)$$

Neighbor Selection

- For a given active user, a, select correlated users to serve as source of predictions.
  - Standard approach is to use the most similar n users, u, based on similarity weights, $w_{a,u}$
  - Alternate approach is to include all users whose similarity weight is above a given threshold. $\mathrm{Sim}(r_a, r_u) > t$



Significance Weighting

- Important not to trust correlations based on very few co-rated items.
- Include *significance weights*, $s_{a,u}$, based on number of co-rated items, m.

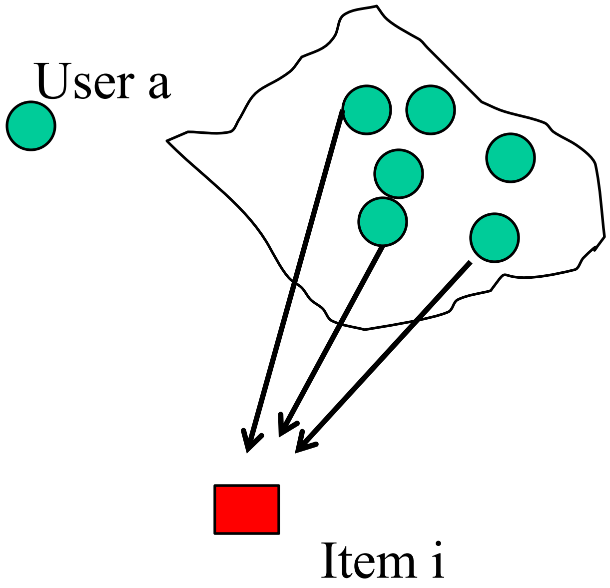
$$w_{a,u} = s_{a,u} c_{a,u}$$

$$s_{a,u} = \begin{Bmatrix} 1 \text{ if } m > 50 \\ \dfrac{m}{50} \text{ if } m \leq 50 \end{Bmatrix}$$

Rating Prediction (Version 0)

- Predict a rating, $p_{a,i}$, for each item i, for active user, a, by using the n selected neighbor users, $u \in \{1,2,\ldots n\}$.
- Weight users' ratings contribution by their similarity to the active user.

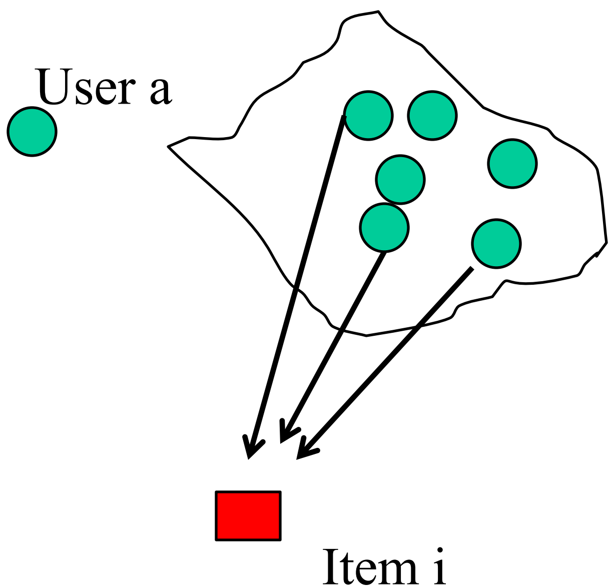
$$p_{a,i} = \frac{\sum_{u=1}^{n} w_{a,u} r_{u,i}}{\sum_{u=1}^{n} w_{a,u}}$$



Rating Prediction (Version 1)

- Predict a rating, $p_{a,i}$, for each item i, for active user, a, by using the n selected neighbor users, $u \in \{1,2,\ldots n\}$.
- To account for users different ratings levels, base predictions on *differences* from a user's *average* rating.
- Weight users' ratings contribution by their similarity to the active user.

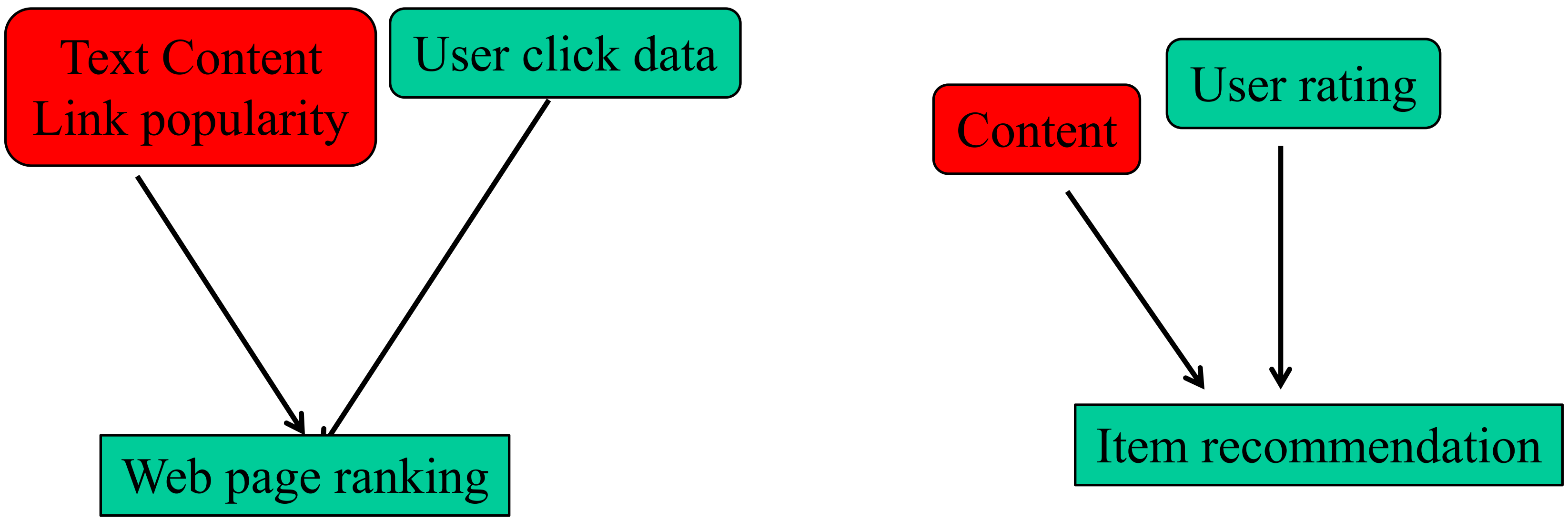
$$p_{a,i} = \bar{r}_a + \frac{\sum_{u=1}^{n} w_{a,u}(r_{u,i} - \bar{r}_u)}{\sum_{u=1}^{n} w_{a,u}}$$



Problems with Collaborative Filtering

- **Cold Start**: There needs to be enough other users already in the system to find a match.
- **Sparsity**: If there are many items to be recommended, even if there are many users, the user/ratings matrix is sparse, and it is hard to find users that have rated the same items.
- **First Rater**: Cannot recommend an item that has not been previously rated.
  - New items, esoteric items
- **Popularity Bias**: Cannot recommend items to someone with unique tastes.
  - Tends to recommend popular items.

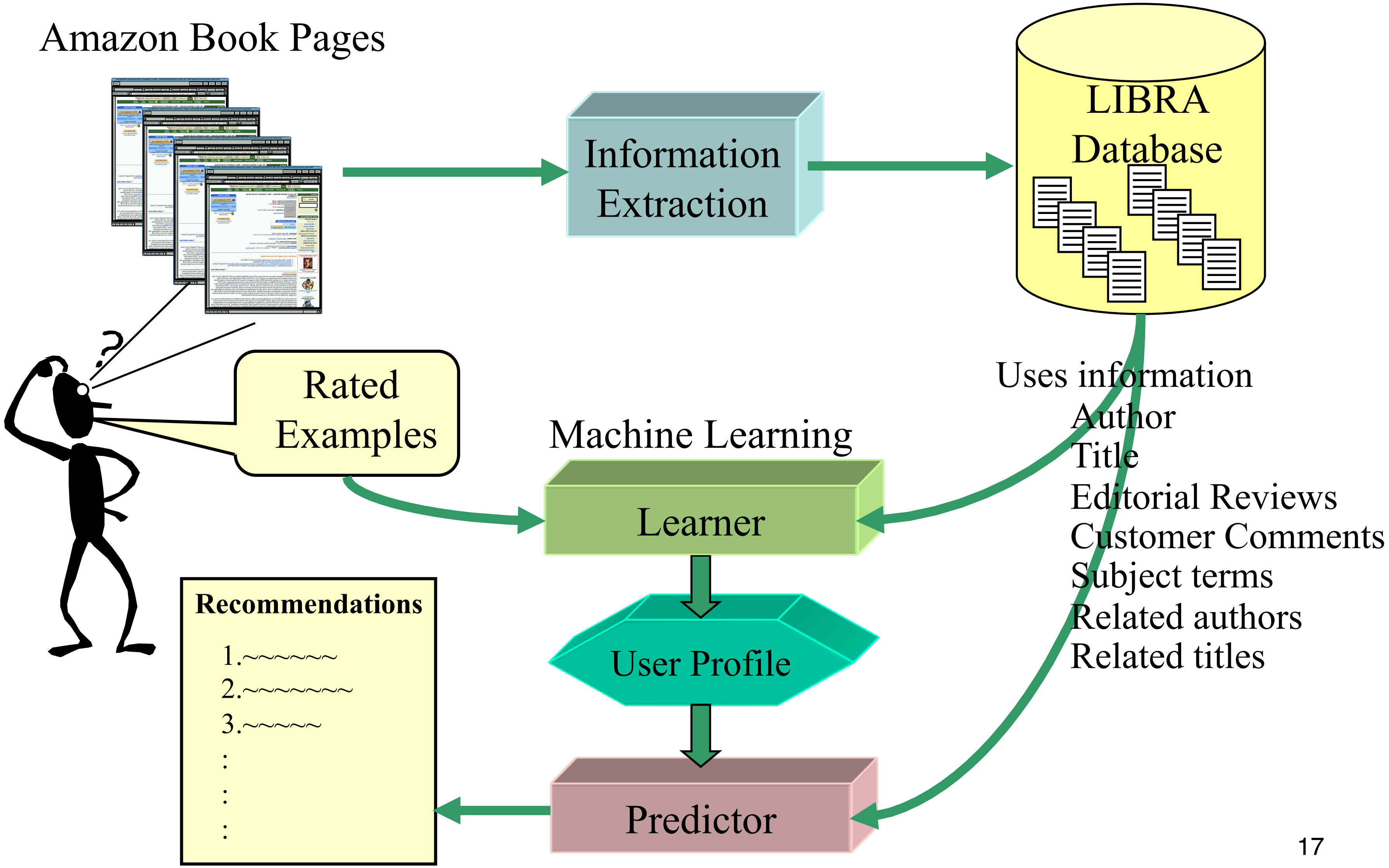
Recommendation vs Web Ranking



Content-Based Recommendation

- Recommendations are based on information on the content of items rather than on other users' opinions.
  - Less dependence for data on other users.
- Able to recommend to users with unique tastes.
- Able to recommend new and unpopular items
  - No first-rater problem.
  - No cold-start or sparsity problems..

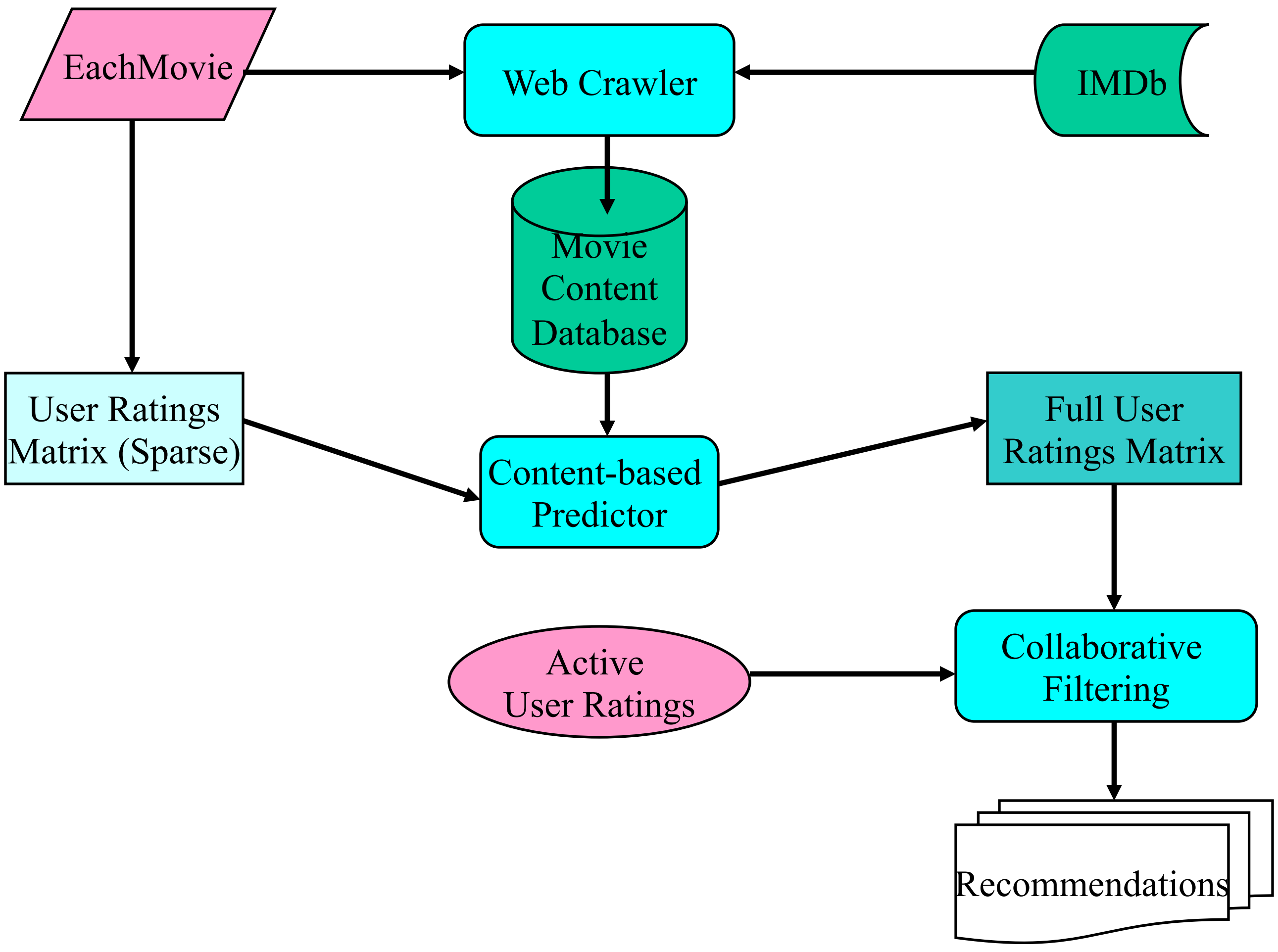
Example: LIBRA System



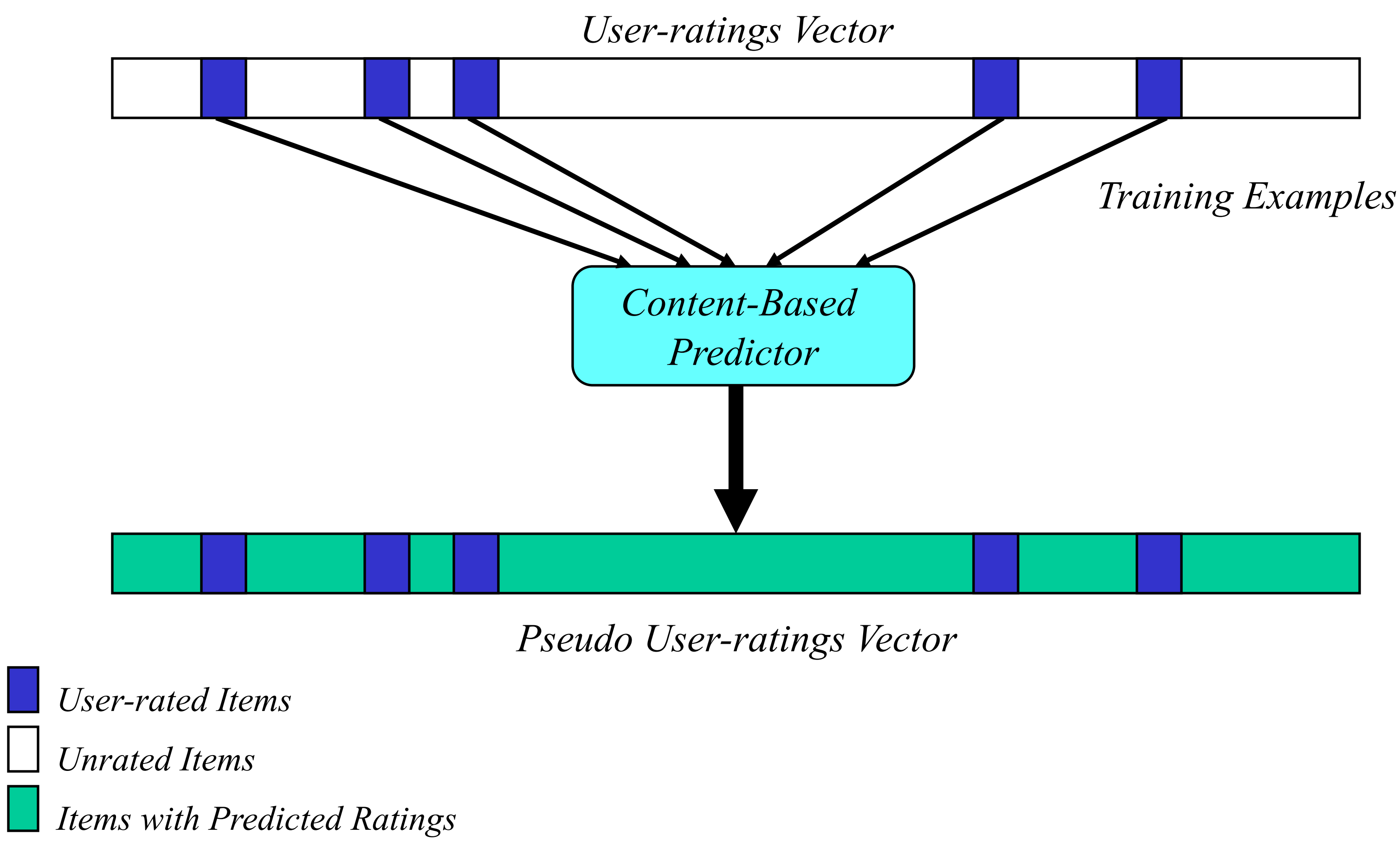
Combining Content and Collaboration

- Content-based and collaborative methods have complementary strengths and weaknesses.
- Combine methods to obtain the best of both.
- Various hybrid approaches:
  - Apply both methods and combine recommendations.
  - Use collaborative data as content.
  - Use content-based predictor as another collaborator.
  - **Use content-based predictor to complete collaborative data.**

Content-Boosted Collaborative Filtering



Content-Boosted Collaborative Filtering



Content-Boosted Collaborative Filtering

- Compute pseudo user ratings matrix
  - Full matrix – approximates actual full user ratings matrix
- Perform collaborative filtering
  - Using Pearson corr. between pseudo user-rating vectors

Conclusions

- Recommending and personalization are important approaches to combating information over-load.
- Machine Learning is an important part of systems for these tasks.
- Collaborative filtering has problems.
- Content-based methods address these problems (but have problems of their own).
- Integrating both is best.