COMP6237 Data Mining

Making Recommendations

Jonathon Hare
jsh2@ecs.soton.ac.uk

Introduction

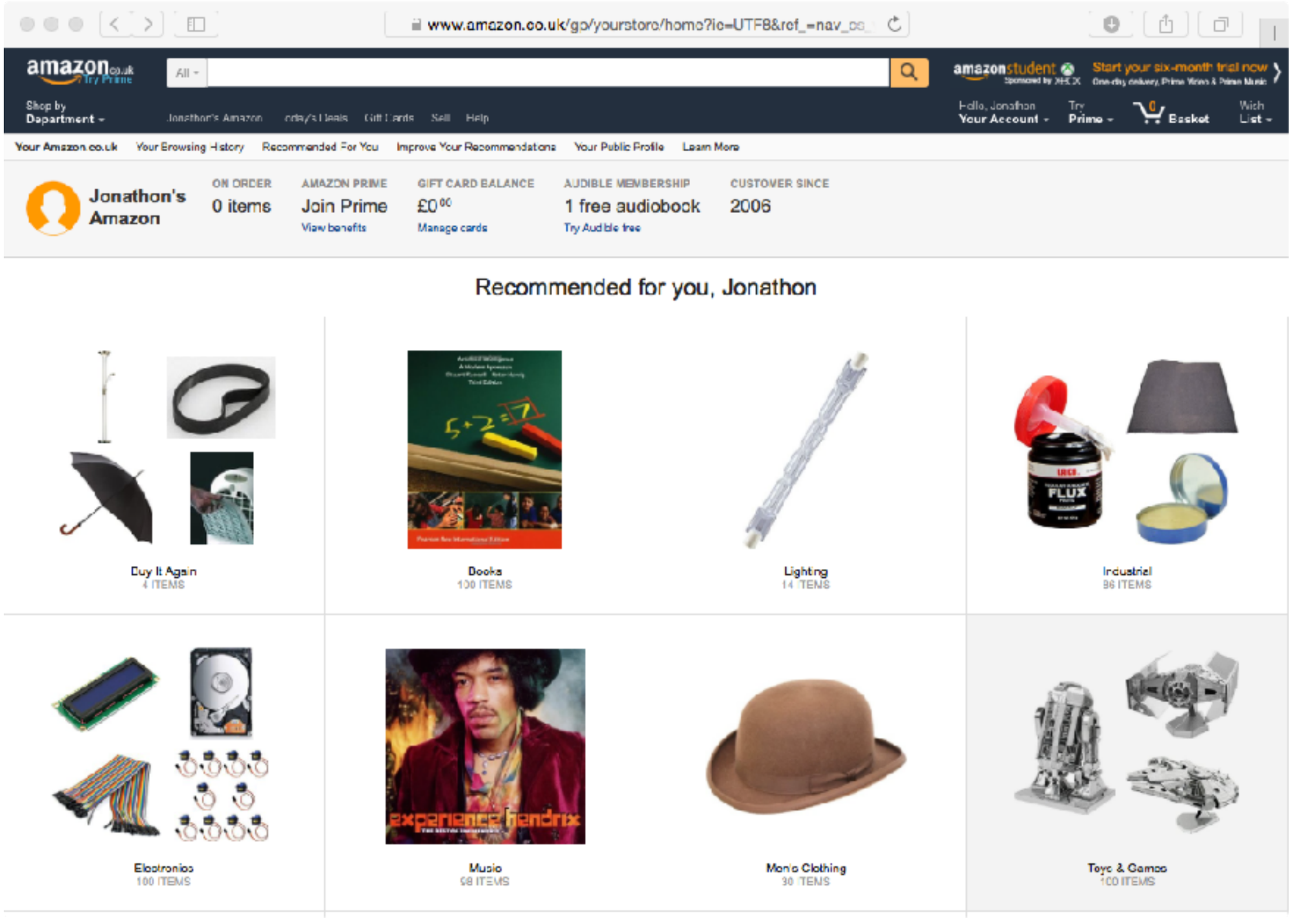
- Recommender systems 101
- Taxonomy of recommender systems
- Collaborative Filtering
  - Collecting user preferences as sparse vectors
  - Finding similar users
  - User and item based filtering

Problem statement: making recommendations

- Can we mine data to predict what
  - things people might like to buy
  - films they would like to watch
  - websites and books they might like to read
  - people other people might like to date
  - …
- on the basis of **past behaviour** and **shared tastes**

Recommender systems 101

Amazon makes recommendations based on past **purchase history**



Web Images Videos Maps **News** Shopping Gmail more ▾

Google news

Search News | Search the Web | Advanced news search | Preferences

Personalized

Recommended for @gmail.com

- Top Stories
- **Recommended**
- U.S.
- World
- Sci/Tech
- Business
- Sports
- Entertainment
- Spotlight
- Health
- Most Popular

› **All news**
Headlines
Images


FOXNews

Obama Nobel Peace Prize: Obama wins, and partisan fighting continues
Chicago Tribune - Mark Z. Barabak, Geraldine Baum - **45 minutes ago**
President Barack Obama's winning of the Nobel Peace Prize brought nothing of the sort at home, as political combatants were quick to assume their usual battlements: Democrats largely hailed the **...**
Video: Did Obama Deserve Nobel Prize? CBS
If Obama can get one, you can, too Detroit Free Press
New York Times - Philadelphia Inquirer - Fort Worth Star Telegram - Wikipedia: 2009 Nobel Peace Prize
all 10,171 news articles » Email this story


Chippewa Herald

US Senate panel votes to extend security law
Reuters - Thomas Ferraro, Anthony Boadle - **Oct 8, 2009**
WASHINGTON (Reuters) - A Senate Judiciary Committee, drawing criticism from both liberals and conservatives, voted on Thursday to extend expiring provisions of a post-September 11 law designed to protect the United States from another attack.
AP Interview: White House expands climate campaign The Associated Press
US Senate Panel Unlikely To Debate CO2 Bill Before Nov Wall Street Journal
New York Times - Houston Chronicle - Politico - Red, Green, and Blue
all 265 news articles » Email this story


Portfolio.com

Barnes & Noble May Sell Its Own E-reader
PC World - Harry McCracken - **Oct 9, 2009**
Is bookstore behemoth Barnes & Noble about to enter the e-book fray with its own Android-powered device? I like these rumors: The Wall Street Journal is reporting that bookstore behemoth Barnes & Noble will soon start selling its own e-reader device, **...**
Barnes & Noble's E-Reader Gets Real Wired News
Barnes & Noble's Sales Down In Aug-Sep; New View Given Wall Street Journal
CNET News - San Francisco Chronicle - Register - FOXNews
all 208 news articles » Email this story


Birmingham Star

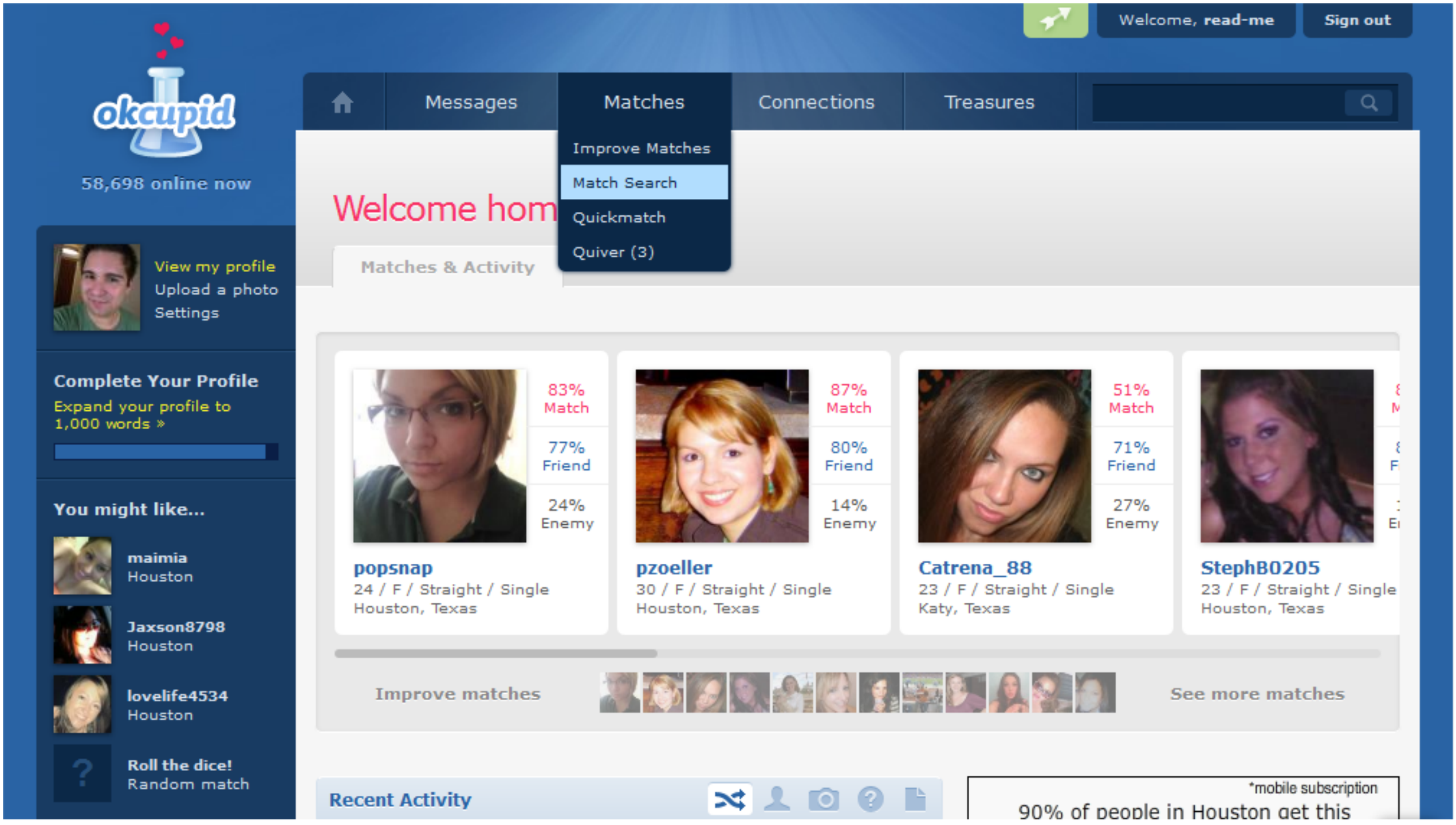
Dow Ends Week at Highest Level in a Year
Washington Post - **3 hours ago**
US stocks gained last week, pushing the Dow Jones industrial average to its highest close in a year, as Alcoa unexpectedly reported a profit and economic data signaled the US recession is ending.
Duo of IBM, Intel Propels Dow's Run Wall Street Journal
Stocks Finish with Gains BusinessWeek
Bloomberg - Reuters - The Associated Press
all 178 news articles » IBM Email this story

Google News makes recommendations based **click and search history**

millions of articles & millions of users

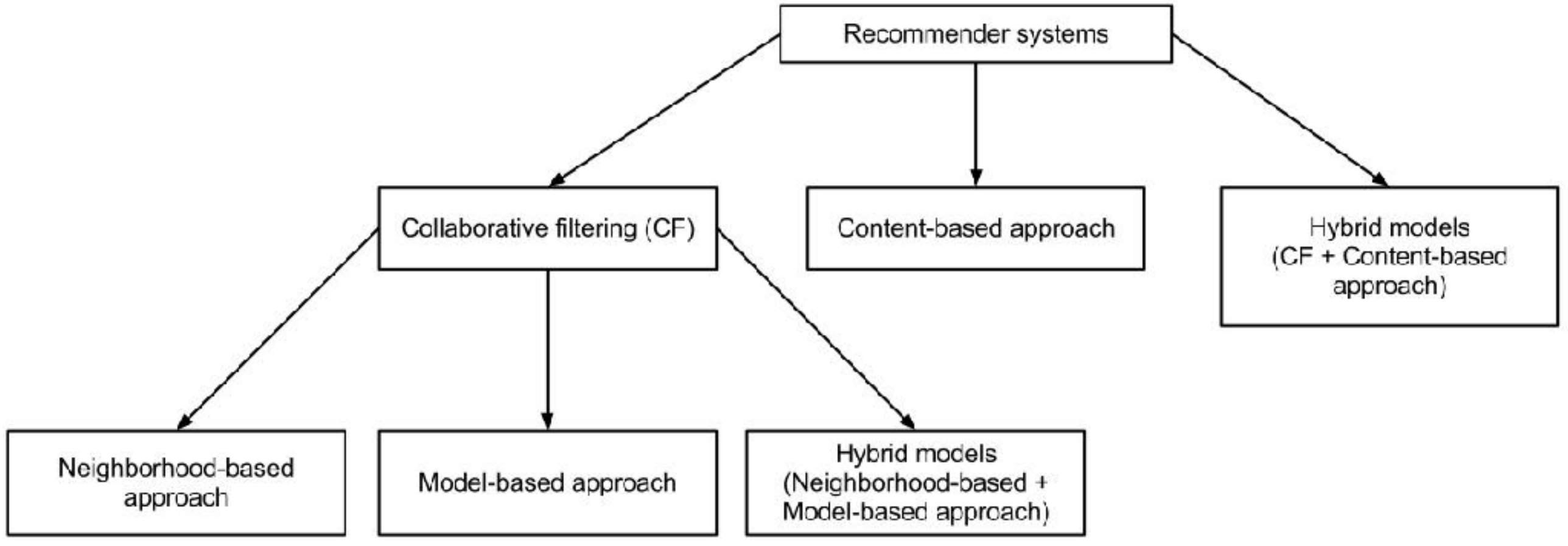
Netflix predicts movies you ♡ based on past **numeric ratings**

okcupid predicts people you might ♡ based on past **site usage/ behaviour** and **answers to questions**



facebook predicts people you might know on your "position" within a **social network** made up of *friends* with different sets of interests, geographical locations, educational backgrounds, …

Recommender systems taxonomy



Collaborative Filtering

Group of users

Group of items

Collaborative Filtering

Group of users

Group of items

Observe user preferences

Collaborative Filtering

Group of users

Group of items

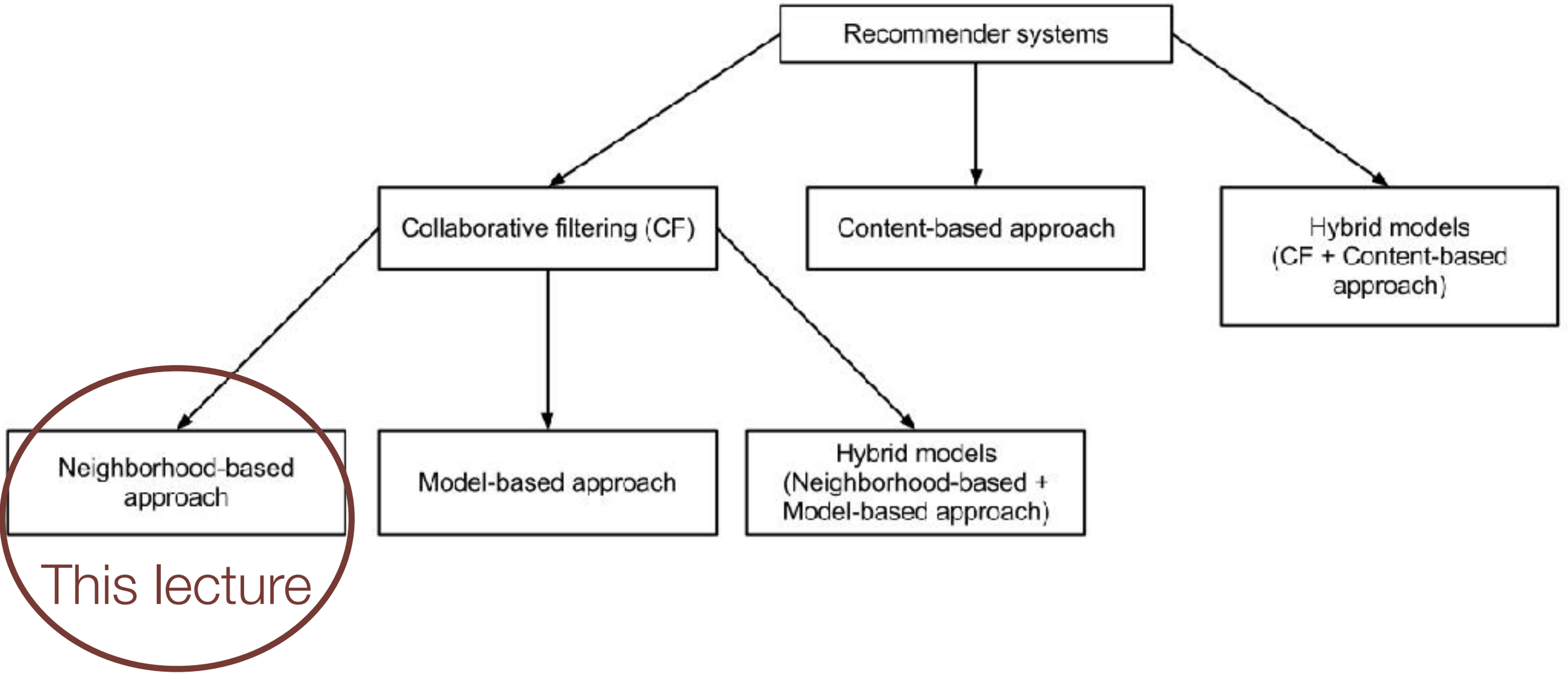
Make predictions about new preferences:

does Bob like strawberries?

Collaborative Filtering and Recommender Systems

- **Insight:** personal preferences are correlated
    - If Jack loves A and B, and Jill loves A, B, and C, then Jack is more likely to love C
- Collaborative Filtering Task
    - Discover patterns in observed preference behaviour (e.g. purchase history, item ratings, click counts) across community of users
    - Predict new preferences based on those patterns
- *Does not rely on item or user attributes (e.g. demographic info, author, genre)*
    - *Content-based recommendation: complementary approach*

Recommender systems taxonomy



Collecting user preferences

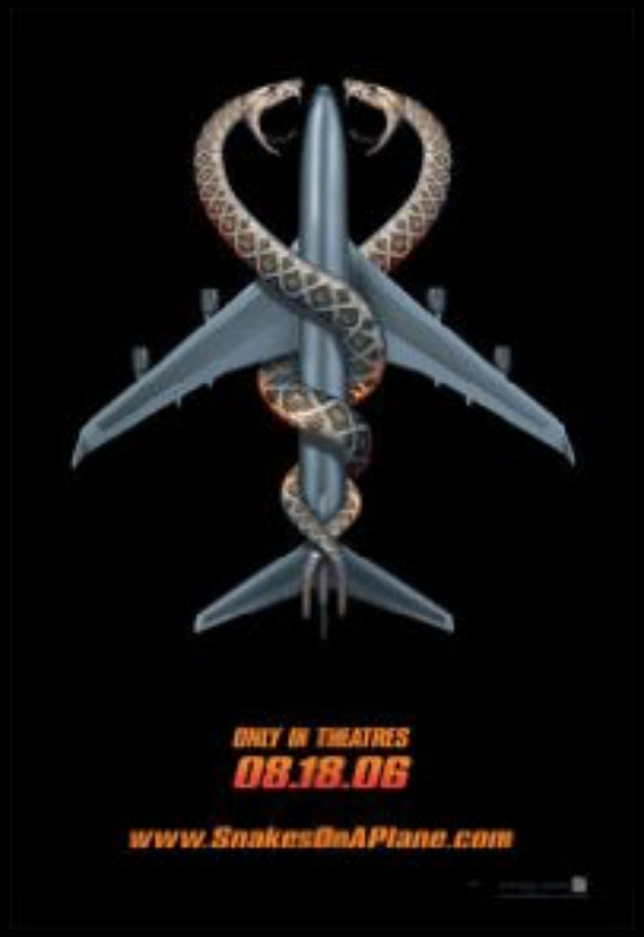
- Need to make measurements of user preferences.
- For example, consider film recommendation:
    - Might ask users to rate films between 0 and 5 stars
    - Rating could be continuous - can allow fractional values, or could be integer

Mapping user actions to numerical scores

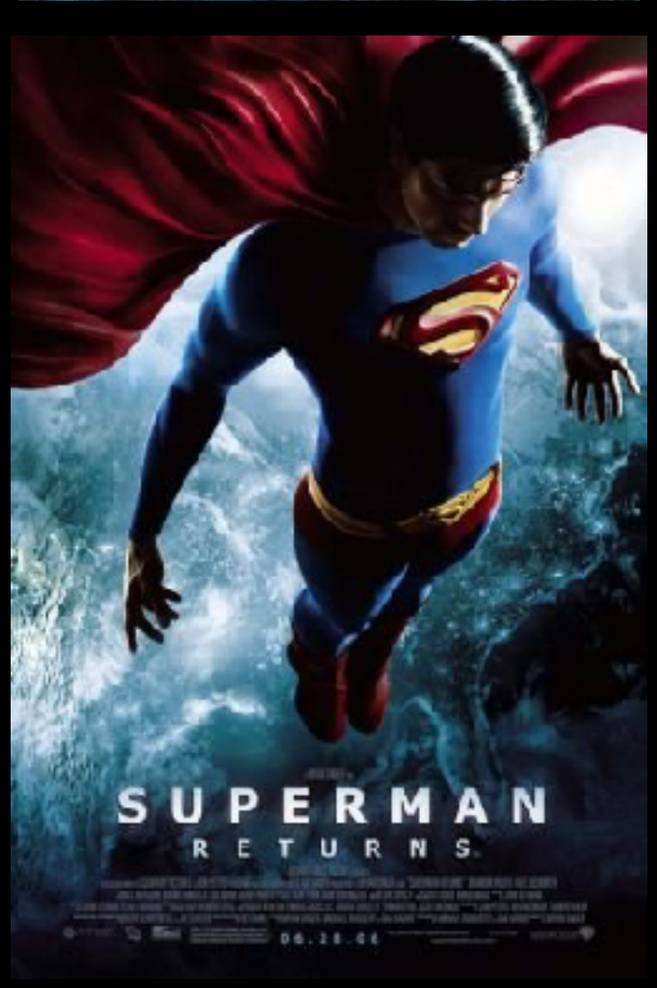
- Many different ways to map user preferences to numerical scores
  - Will depend on the variables that can be measured
- For example:

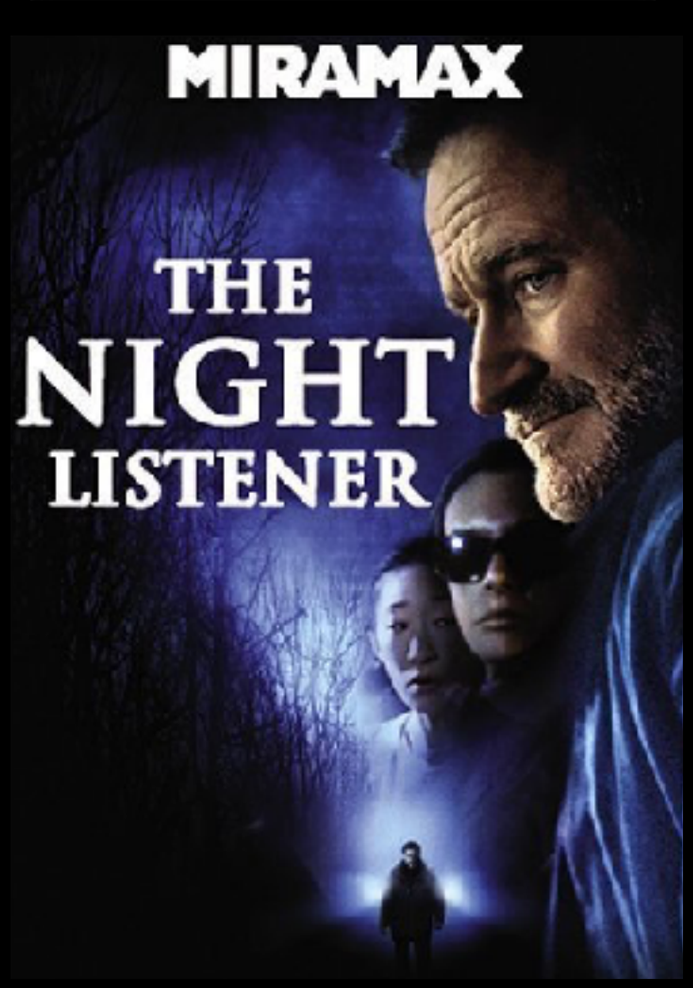
Concert Tickets		Online Shopping		Site Recommender	
Brought	1	Brought	2	Liked	1
Didn't Buy	0	Browsed	1	No vote	0
		Didn't buy	0	Disliked	-1


A FILM BY
M. NIGHT SHYAMALAN
PAUL GIAMATTI
BRYCE DALLAS HOWARD
LADY IN THE WATER
JULY 21


ONLY IN THEATRES
08.18.06
www.SnakesOnAPlane.com


From the director of 'How to Lose a Guy in 10 Days.'
Lindsay Lohan
Chris Pine
Everything changed in the wink of an eye.
Just My Luck


SUPERMAN
RETURNS
06.28.06


MIRAMAX
THE NIGHT LISTENER


Owen Wilson
Kate Hudson
Matt Dillon
Michael Douglas
You, Me and Dupree
July 14
Two's company. Dupree's a crowd.

Example dataset:

	Lady in the Water	Snakes on a Plane	Just My Luck	Superman Returns	The Night Listener	You, Me and Dupree
Lisa	2.5	3.5	3.0	3.5	3.0	2.5
Gene	3.0	3.5	1.5	5.0	3.0	3.5
Michael	2.5	3.0		3.5	4.0	
Claudia		3.5	3.0	4.0	4.5	2.5
Mick	3.0	4.0	2.0	3.0	3.0	2.0
Jack	3.0	4.0		5.0	3.0	3.5
Toby		4.5		4.0		1.0

This dataset is *sparse.* It has missing values!

	Lady in the Water	Snakes on a Plane	Just My Luck	Superman Returns	The Night Listener	You, Me and Dupree
Lisa	2.5	3.5	3.0	3.5	3.0	2.5
Gene	3.0	3.5	1.5	5.0	3.0	3.5
Michael	2.5	3.0		3.5	4.0	
Claudia		3.5	3.0	4.0	4.5	2.5
Mick	3.0	4.0	2.0	3.0	3.0	2.0
Jack	3.0	4.0		5.0	3.0	3.5
Toby		4.5		4.0		1.0

Aside: Sparse data

- This sparsity is important
    - We can use it to our advantage:
        - More efficient storage
        - Faster computation
- But we have to use appropriate data structures
    - Sparse vectors
        - Typically implemented with hash maps (fast random insertion) or parallel sorted arrays (slow random insertion but faster reads)
- It has disadvantages too…

Aside: spaces, distances and similarity

DATA
(input)

Preprocessing

Feature Extraction

Intelligent Algorithm

Data Mining

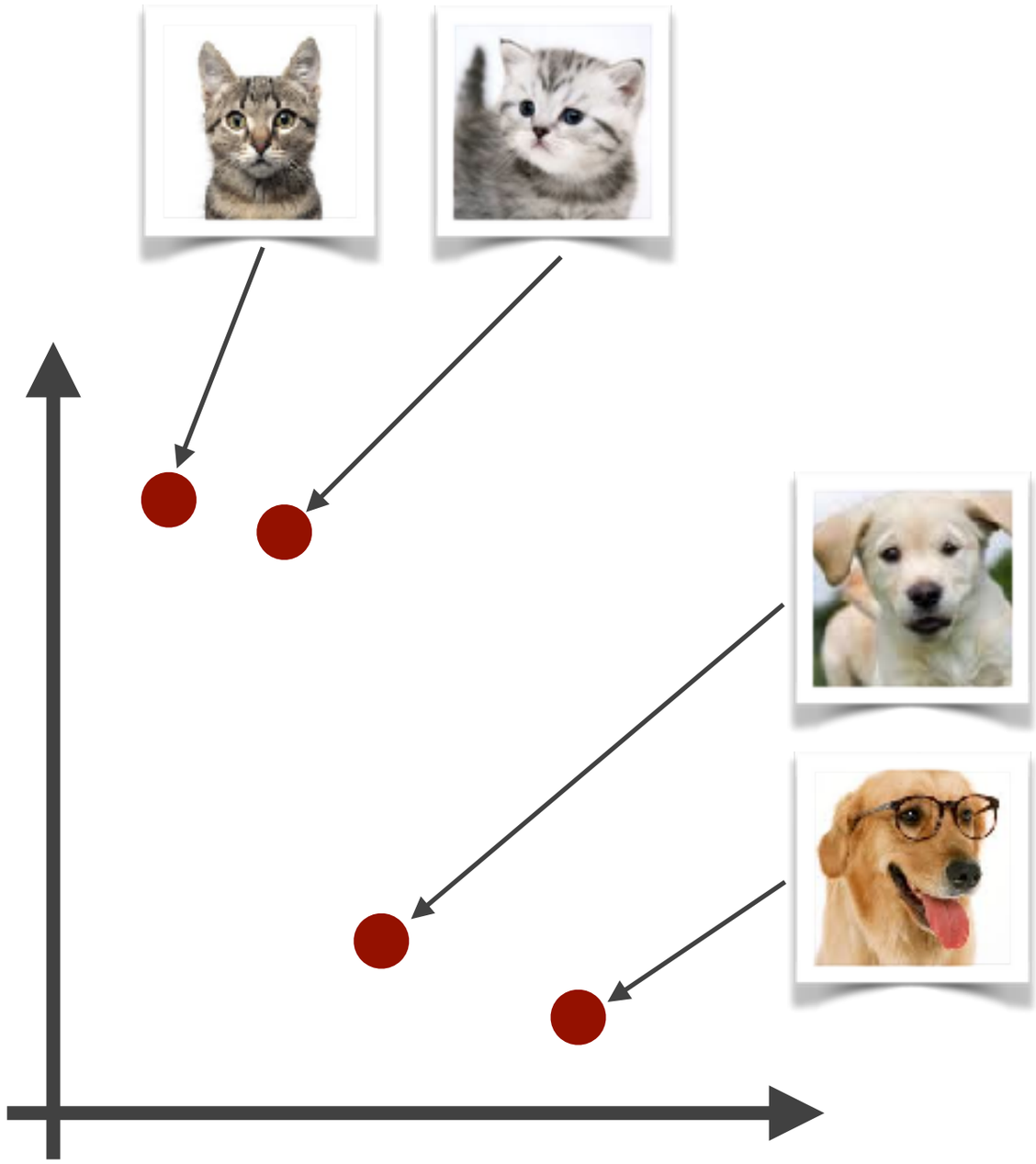
INFORMATION
(output)

Key terminology

- **featurevector**: a mathematical vector
  - just a list of (*usually Real*) numbers
  - has a fixed number of **elements** in it
    - The number of elements is the **dimensionality** of the vector
- represents a **point** in a **featurespace** or equally a **direction** in the featurespace
- the **dimensionality of a featurespace** is the dimensionality of every vector within it
  - vectors of differing dimensionality can’t exist in the same feature space

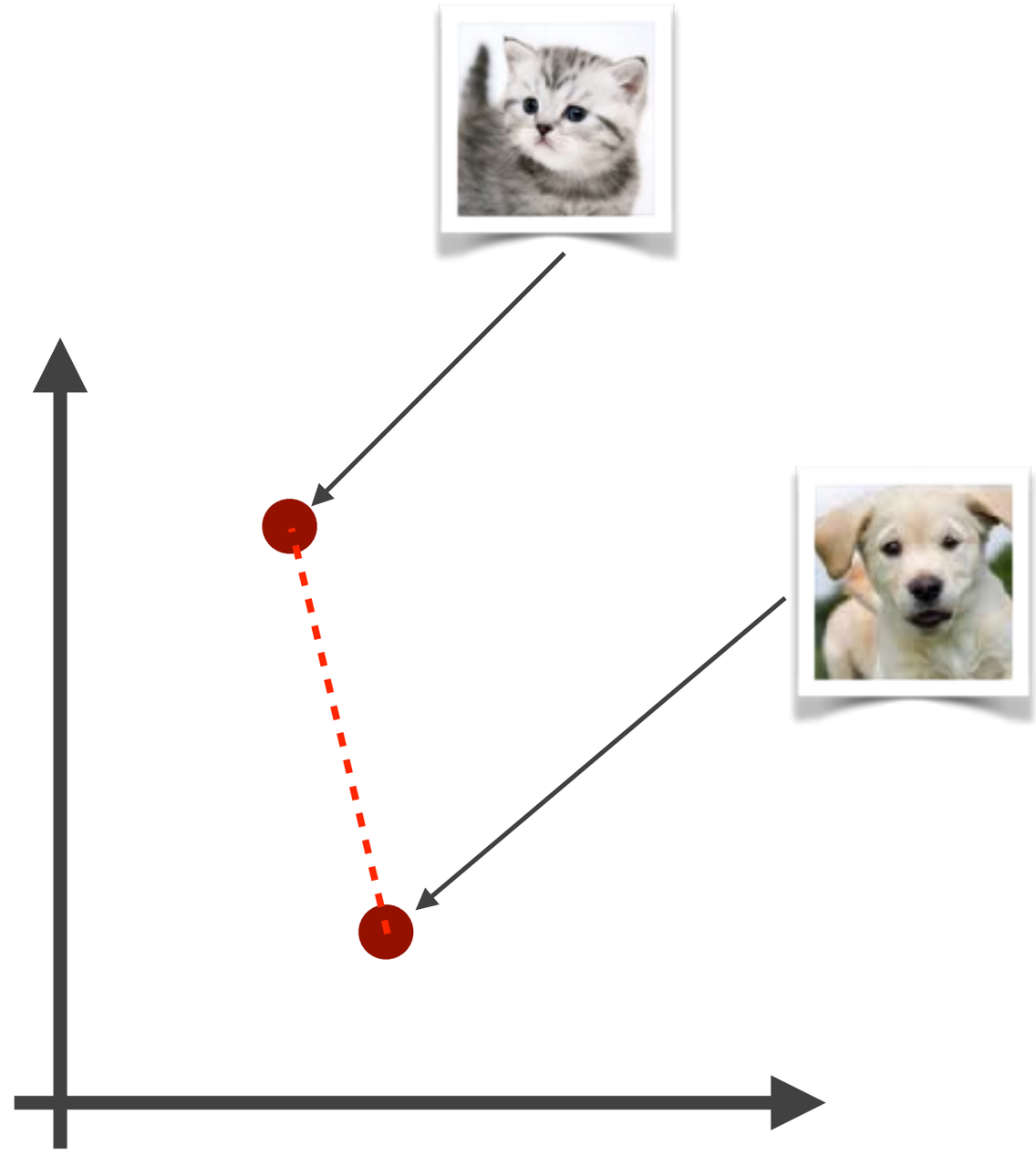
Distance and similarity

- Feature extractors are often defined so that they produce vectors that are *close* together for *similar* inputs
  - Closeness of two vectors can be computed in the feature space by measuring a distance between the vectors.



Euclidean distance (*L2 distance*)

- L2 distance is the most intuitive distance…
  - The straight-line distance between two points
  - Computed via an extension of Pythagoras theorem to *n* dimensions:

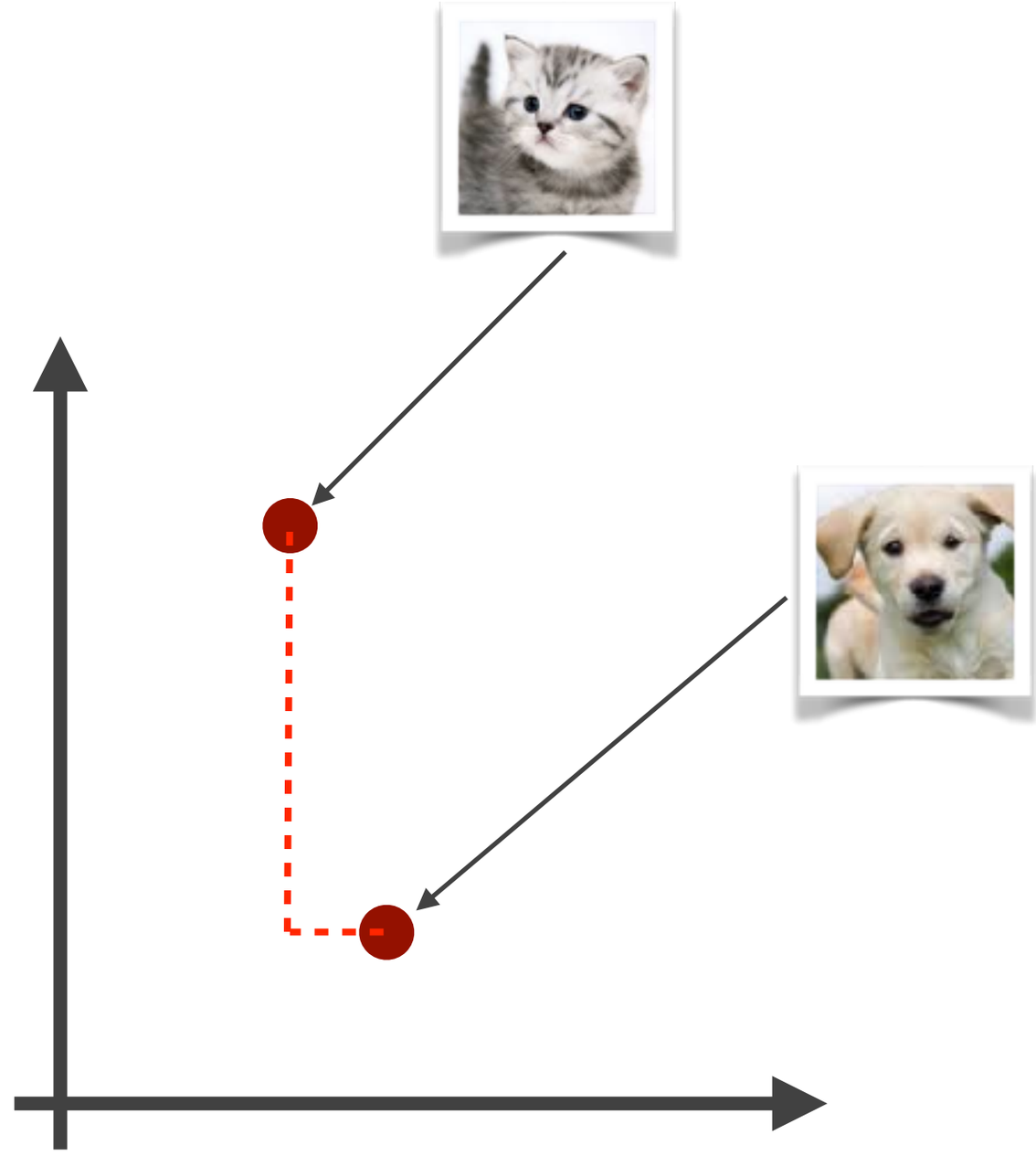


$$\mathrm{D}_2(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2} = ||\mathbf{p}-\mathbf{q}|| = \sqrt{(\mathbf{p}-\mathbf{q})\cdot(\mathbf{p}-\mathbf{q})}$$

L1 distance (*aka Taxicab/Manhattan*)

- L1 distance is computed along paths parallel to the axes of the space:

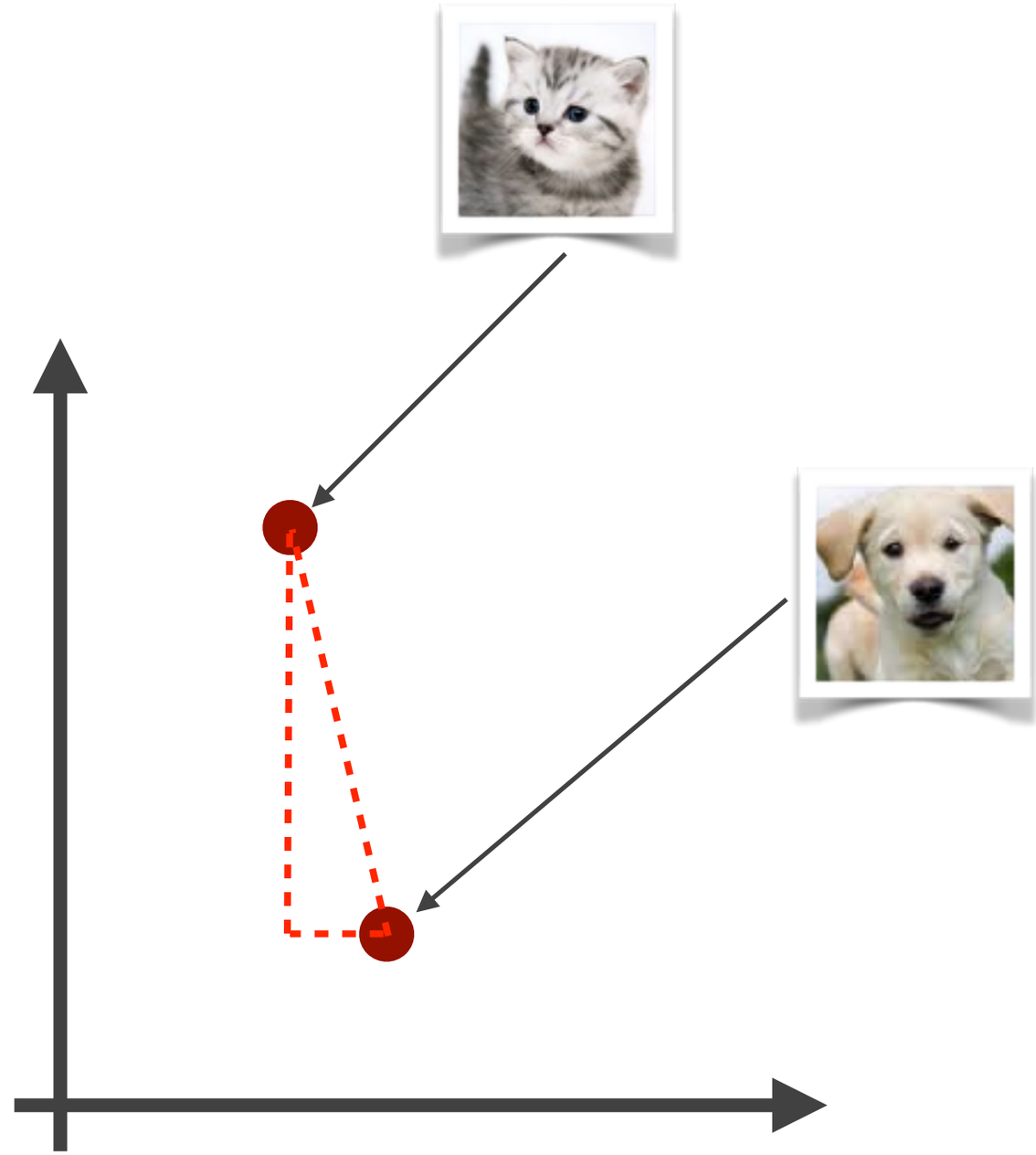
$$\mathrm{D}_1(\mathbf{p}, \mathbf{q}) = ||\mathbf{p} - \mathbf{q}||_1 = \sum_{i=1}^{n} |p_i - q_i|$$



The Lp distances

- Generalisation of the L1 and L2 distances to higher orders:

$$\mathrm{D}_p(\mathbf{x}, \mathbf{y}) = ||\mathbf{x} - \mathbf{y}||_p = (\sum_{i=1}^{n} |x_i - y_i|^p)^{\frac{1}{p}}$$

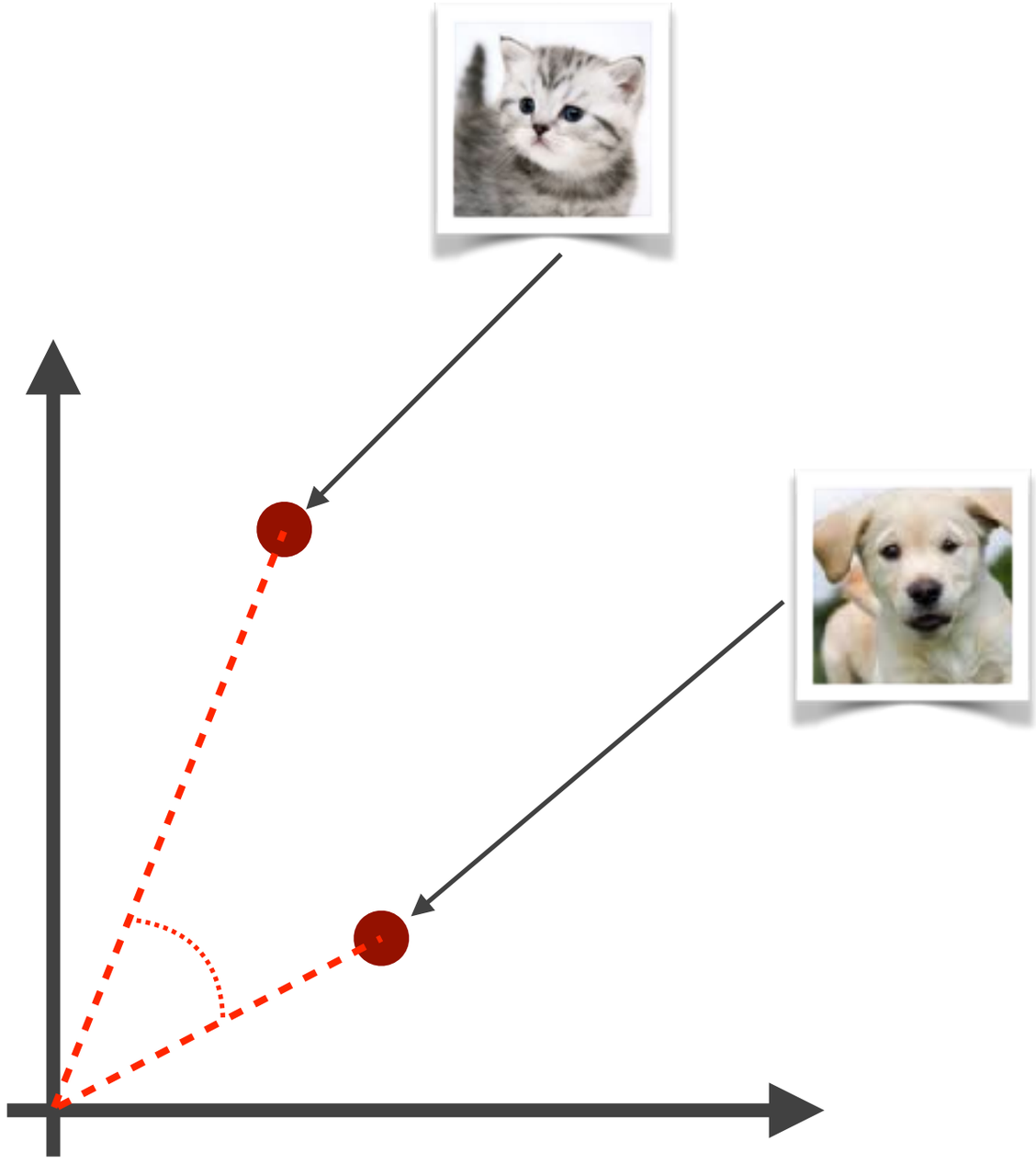


Cosine Similarity

- Cosine similarity measures the cosine of the angle between two vectors
  - **It is not a distance!**

$$\cos(\theta) = \frac{p \cdot q}{||p||||q||} = \frac{\sum_{i=1}^{n} p_i q_i}{\sqrt{\sum_{i=1}^{n} p_i^2}\sqrt{\sum_{i=1}^{n} q_i^2}}$$

- Useful if you don't care about the relative length of the vectors



Measuring user similarity

- Can we use this data to measure and rank the similarity of users?
  - Need to define a “similarity measure” or “similarity score”
    - On the basis that similar users have similar tastes (i.e. like the same movies)
  - **Must take into account sparsity - not all users have seen all movies**
  - *Typically* the score is a bounded numeric value
    - 1 -> the same …. 0 -> completely dissimilar

Demo: Plots of Euclidean Space

Euclidean Similarity

- Many ways to compute a “similarity” based on Euclidean distance
  - We could choose:

$$\text{sim}_{L2}(x, y) = \frac{1}{1 + \sqrt{\sum_{i \in I_{xy}} (r_{x,i} - r_{y,i})^2}}$$

where $r_{X,i}$ refers to the rating given by user *x* to item *i*, and I_{xy} is the set of items rated by both user *x* and *y*

Pearson Correlation

- An alternative measure of similarity is to compute the correlation of the users based on the ratings they share
  - Pearson's correlation is the standard measure of dependence between two r.v's:

$$\text{sim}_{pearson}(x, y) = \frac{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r_x})(r_{y,i} - \bar{r_y})}{\sqrt{\sum_{i \in I_{xy}} (r_{x,i} - \bar{r_x})^2 \sum_{i \in I_{xy}} (r_{y,i} - \bar{r_y})^2}}$$

where $\bar{r}_x$ is the average rating user x gave for all items in I_{xy}

Demo: Plots of Correlation

Important Aside: Grade inflation

- Users are notoriously bad at giving consistent absolute ratings
    - two users might agree that they both enjoyed a film, but one might give it a 4 and the other a 5
        - e.g. compare Lisa & Jack in the sample data
- Pearson correlation corrects for this automatically, but the Euclidean similarity doesn't
    - Data normalisation and mean centring can overcome this
        - data *standardisation*

User-based Filtering: Rating and Ranking the critics

- We now have a set of measures for computing the similarity between users.

- Can use this to produce a *ranked list* of the best matches (most similar users) to a target user

- Typically want the *top-N* users

- When computing the ranked list, might only want to consider a subset of users

  - e.g. those who rated a particular item

Demo: User-User similarity

User-based Filtering: Recommending Items

- Now we know how to find similar users, how can we recommend items?

- Basic idea:

- predict the rating, $r_{u,i}$, of an item *i* by user *u* as an aggregation of the ratings of item *i* by users similar to *u:*

$$r_{u,i} = \text{aggr}_{\hat{u} \in U}(r_{\hat{u},i})$$

where *U* is the set of *top* users most similar to *u* that rated item *i.*

One possible aggregation approach:

- Weight the scores of each item each similar user has rated
  - i.e. multiply the item scores by the respective users' similarity
- and combine into a score for each item
- Problem: can't just add them together - items that have more ratings would always have an advantage
  - Need to normalise by the sum of ratings:

$$r_{u,i} = \frac{\sum_{\hat{u} \in U} \text{sim}(u, \hat{u}) r_{\hat{u},i}}{\sum_{\hat{u} \in U} |\text{sim}(u, \hat{u})|}$$

Demo: User-based recommendation

(Some) other aggregation options

$$r_{u,i} = \frac{1}{N} \sum_{\hat{u} \in U} r_{\hat{u},i}$$

don't weight by similarity score (but still only compute the average over similar users)

$$r_{u,i} = \bar{r}_u + \frac{\sum_{\hat{u} \in U} \text{sim}(u, \hat{u})(r_{\hat{u},i} - \bar{r}_{\hat{u}})}{\sum_{\hat{u} \in U} |\text{sim}(u, \hat{u})|}$$

$\bar{r}_u$ is the average score of user *u* over all items they have scored. This aggregation compensates for users that have high or low averages.

Matching Products:“More like this”

- In just the same way that we computed similarity between users, we could flip (**transpose**) the problem and compute the similarity between items
    - Can use as a *fuzzy* basis for recommending alternative items
- In a future lecture we’ll look at a more structured way of identifying what products people buy together using “affinity analysis” or “Market Basket Analysis”

Demo: Item-Item similarity

Problems with user based filtering

- User-based filtering relies on computing the similarity against every user
    - With millions of users this might be a problem!
        - Computationally hard
        - If there are thousands of products there might be little overlap between users…
            - …making effective similarity computation hard

Item based collaborative filtering

- Designed to work-around problems of user based filtering
    - based on the idea that *comparisons between items will not change as frequently as comparisons between users*
- Steps:
    - Precompute the most similar items for each item and store them
    - To make a recommendation for a user, look at their top-rated items and aggregate items similar to those from the precomputed item similarities
- The precomputed item similarities will obviously change with new ratings, but they will do so **slowly**

Demo: Precomputing Item-Item similarity

Computing recommendations

- For an unrated item (by user *u*), *î*, that has a top-N similarity to an item *i*, that the user has rated, estimate the rating as

$$r_{u,\hat{i}} = \frac{\sum_{i \in I} \mathrm{sim}(\hat{i}, i) r_{u,i}}{\sum_{i \in I} \mathrm{sim}(\hat{i}, i)}$$

where *I* is the subset of all N items similar to *î*

Demo: Item-based recommendation

Tradeoffs between user-based and item-based filtering

- User-based filtering is easier to implement & doesn't have the overhead of maintaining the item-item comparisons

    - User-based filtering also deals better with datasets that frequently change

- For small, relatively dense datasets both user-based and item-based filtering perform equally well

- For large, sparse datasets item-based filtering tends to work better

The “cold-start” problem

- What happens when a new user signs up to use our recommendation service or a new item is added?
    - CF won’t work if we don’t have any ratings for the new user/item
- This is known as the “**cold-start**” problem

Potential solutions for new items

- Adopt a hybrid recommendation approach:
  - Use content-based features to find similar items
  - Bootstrap ratings for the new item for the users that rated similar items by averaging the ratings those users gave to similar items

Potential solutions for new users

- Somewhat harder problem
  - Typically need to “bootstrap” the user profile
    - Perhaps use web browsing behaviour (i.e. from *tracking cookies*) to attempt to predict what kinds of things the user is interested in
  - Perhaps ask new users to perform tasks
    - e.g. answer some questions, etc

AudioScrobbler

An ECS Success Story

AudioScrobbler: An ECS Success Story

- Richard Jones developed **Audioscrobbler** as his third year project
  - Recommended new music based on your listening
  - **CF based approach** like we looked at today
- Richard moved to London and launched Audioscrobbler as a Web startup.
  - In 2005 Audioscrobbler.com merged with an internet radio station creating Last.fm
  - CBS Interactive acquired Last.fm for $280 million

UNIVERSITY OF SOUTHAMPTON

Faculty of Engineering and Applied Science

Department of Electronics and Computer Science

A project report submitted for the award of

BSc Computer Science Hons.

Supervisor: Dr. Paul Garrett

Examiner: Dr. JJ Stefanov

Audioscrobbler: Real-time Data Harvesting and Musical Collaborative Filtering

by Richard Jones

May 8, 2003

Join • Lo

last.fm BETA

Search

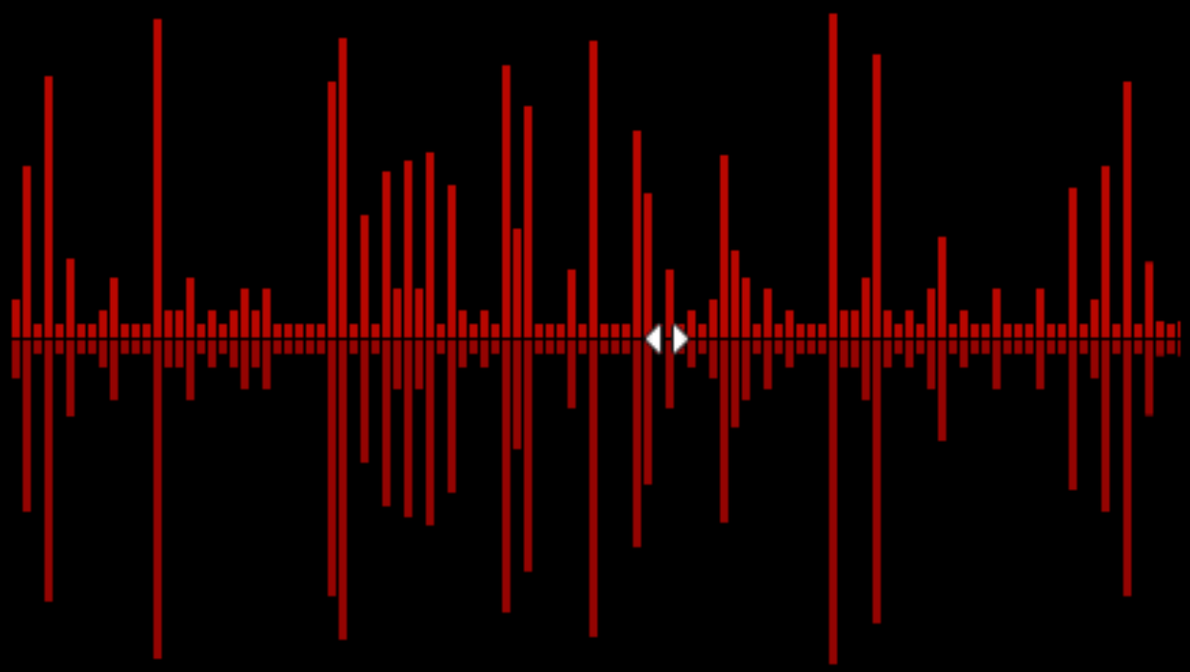
Live Music Events Featur

Hello & welcome to Last.fm.

We bring together your favourite music services and join up listening, watching and sharing to connect your musical world.
Below you can visualise, in real-time, the listening habits & trends of Last.fm's global community. Go Explore.

Scrobbling Now

We're entirely powered by our community of music lovers
Scroll through to see what's being listened to right now



Drag to browse recent scrobbles

Venn Howard

Want to hear some 80s Electronic? How about some millennial Metal
Drag the circles to explore various sounds from different eras

Beastie Boys - Lighten Up

Plays: 363,

Summary

- The ability to make recommendations based on data is one of the big drivers of data mining.
    - Potential to be worth $$$ in numerous industries .
- **Collaborative Filtering** systems make use of user behaviour or ratings in order to gather information to make recommendations.
    - CF systems don’t use content-based features (but other forms for recommender systems do).
    - User-based neighbourhood approach to CF computes similarity between users, and uses this to predict unseen item weights on the basis of aggregations of the ratings of the similar users.