Music Information Retrieval: Datasets and Evaluation

Markus Schedl

http://www.cp.jku.at

JOHANNES KEPLER
UNIVERSITY LINZ

Department of
Computational
Perception

Overview

Music Information Retrieval: A very brief intro

Datasets for music recommendation

- Motivation and context
- Important datasets
- LFM-1b in depth
  - Data acquisition and dataset content
  - Statistical analysis
  - Use case: music recommender systems

Evaluation

- Standard retrieval, machine learning, and rec. sys. metrics
- User-centric measures/aspects

Music Information Retrieval: A very brief introduction

Definitions of Music Information Retrieval

"MIR is a **multidisciplinary** research endeavor that strives to develop innovative **content-based searching schemes**, novel **interfaces**, and evolving **networked delivery** mechanisms in an effort to make the world's vast store of music accessible to all."

(Downie, 2004)

"...actions, methods and procedures for **recovering stored data** to provide information on music."

(Fingerhut, 2004)

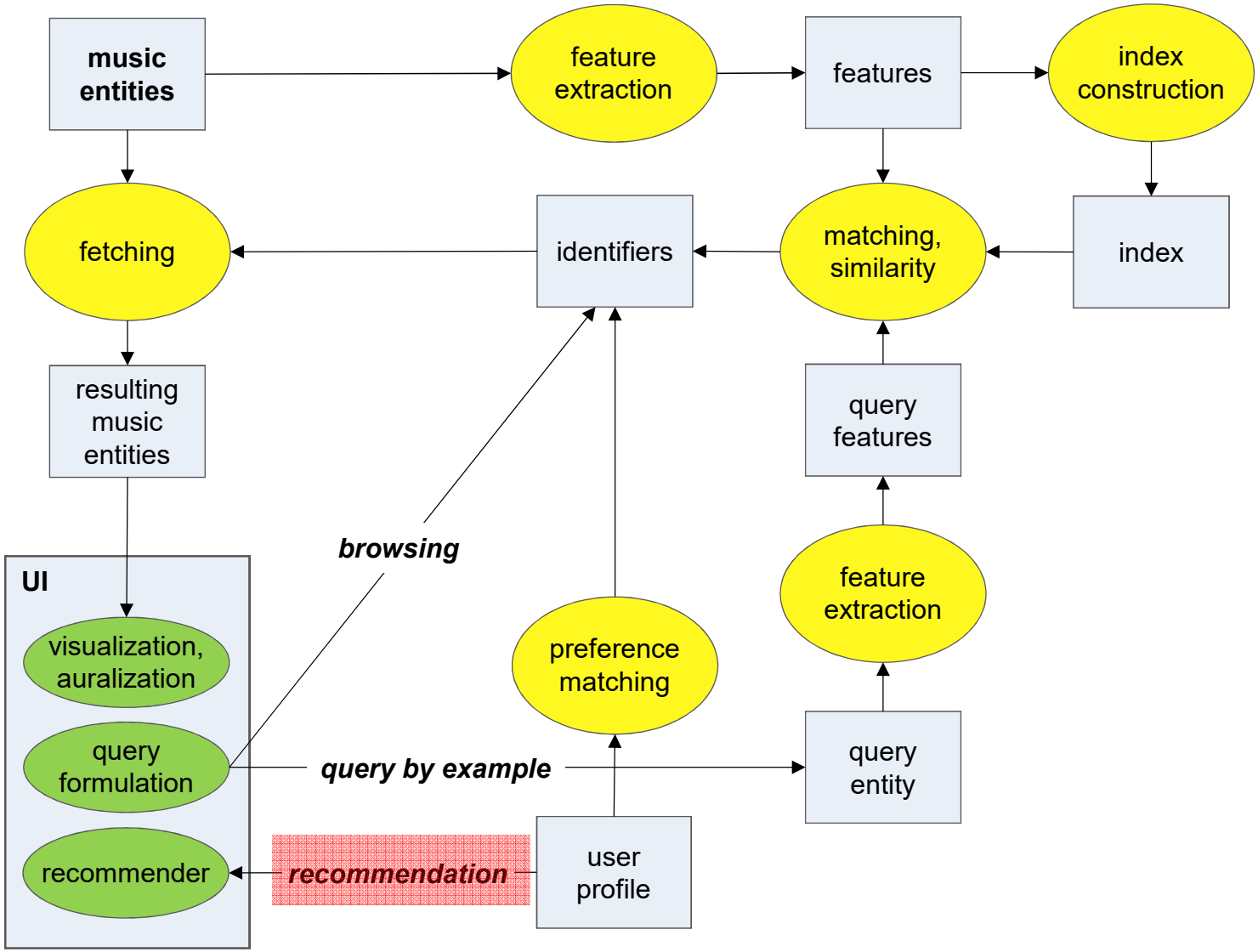
"MIR is concerned with the **extraction**, **analysis**, and **usage** of information about **any kind of music entity** (for example, a song or a music artist) on **any representation level** (for example, audio signal, symbolic MIDI representation of a piece of music, or name of a music artist).

(Schedl, 2008)

Typical MIR Tasks

- Music identification, fingerprinting
- Music alignment (e.g. audio-to-score or audio-to-lyrics)
- Cover song identification
- Query by example: query by humming, query by tapping
- Semantic/tag-based retrieval
- Music recommendation: accuracy, diversity, familiarity, transparency, serendipity
- Music playlist generation or “serial recommendation” (order of tracks important)
- Music browsing interfaces: visualization and auralization
- Comparative performance analysis
- Creative applications

Schematic overview of MIR systems



Datasets for music recommendation

Motivation and context

- Online music services such as music streaming (Spotify, Pandora, …) have led to tens of millions of pieces being available to users easily
- However, (academic) researchers who want to evaluate their MIR approaches on large scale/real world datasets of music metadata typically fetch or crawl these datasets on their own via APIs (Last.fm, Soundcloud, …)

  → result of API calls typically not stable over time

  → fetching large amount of data is time-consuming

  → noisy metadata often requires laborious text processing

  → harms reproducibility

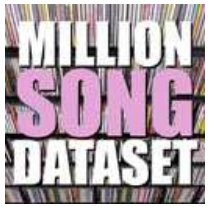
Need for standardized, large-scale datasets!

Datasets: Yahoo! Music

[Dror et al., JMLR, 2012]

- Largest public music ratings dataset:
  - 262M ratings
  - 625M items (tracks, albums, artists, genres)
  - 1M users
- Covering time range 1999–2010
- Used in "KDD Cup 2011" (rating prediction and classification of loved vs. never rated songs), 2000+ participants

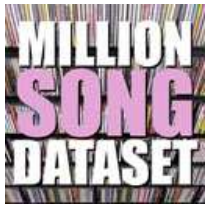
- No user data, no music metadata → limits usage to rating prediction/collaborative filtering (CF) tasks
- Very high sparsity: 99.96% (Netflix: 98.82%)



Datasets: Million Song Dataset

[Bertin-Mahieux et al., ISMIR, 2011]

- Great amount of various music metadata for 1M songs:
  - Content descriptors (key, tempo, loudness, etc.)
  - Editorial metadata
  - Links to MusicBrainz and 7digital
  - Tags and similarity information from Last.fm
  - VSM representations of song lyrics from musiXmatch
  - Information about cover songs
  - Some playcount information ("taste profiles")
- Frequently used in MIR
- "MSD Challenge" in 2012 (rating prediction with variety of sources, http://labrosa.ee.columbia.edu/millionsong/challenge)
- Download: http://labrosa.ee.columbia.edu/millionsong



Datasets: Million Song Dataset

[Bertin-Mahieux et al., ISMIR, 2011]

- Lack of audio data
- Unclear which approaches used for feature extraction
- Integration of (quite heterogeneous) data sources needs improvement

Datasets: Million Musical Tweets Dataset

[Hauger et al., ISMIR, 2013]

- 1M listening events with geo-tags (GPS coordinates) extracted from microblogs:
  - 215K Twitter users
  - 25K artists
  - 1.1M listening events (artist, song, user, time, position)
  - Links to MusicBrainz, 7digital, Amazon
- Extensive temporal data (date, time, weekday, timezone) and spatial data (longitude, latitude, continent, country, county, state, city)
- Download: http://www.cp.jku.at/datasets/MMTD

- Bias towards Twitter users
- Uneven geographical distribution
- Highly varying levels of listening activity between users

Datasets: Last.fm 1k and Last.fm 360k

[Celma, 2010]

- Names, MusicBrainz-IDs, and playcounts of artists most frequently listened to by 360K Last.fm users
- Full Last.fm listening histories for 1K users (user, time stamp, artist, song, MB-IDs)
- Download: http://ocelma.net/MusicRecommendationDataset

- Relatively small
- Covers only up to spring 2009

Summary of datasets

Data Set/Items	Songs	Albums	Artists	Users	Ratings/Evts.
Yahoo! Music [45]	——	624,961 in total	——	1,000,990	262,810,175
MSD [14]	1,000,000			1,019,318	48,373,586
Last.fm – 360K [31]			186,642	359,347	
Last.fm – 1K [31]			107,528	992	19,150,868
MusicMicro [119]	71,410		19,529	136,866	594,306
MMTD [58]	133,968		25,060	215,375	1,086,808
AotM-2011 [88]	98,359		17,332	16,204	859,449

[Schedl et al., 2016]

Summary of datasets

Data Set	Feedback type	Audio files	Item content	User context
Yahoo! Music [45]	ratings	✗	✗	✗
MSD [14]	listening events, tags	✗	✓	✗
Last.fm – 360K [31]	listening events	✗	✓	✗
Last.fm – 1K [31]	listening events	✗	✓	✓
MusicMicro [119]	listening events	✗	✓	✓
MMTD [58]	listening events	✗	✓	✓
AotM-2011 [88]	playlists	✗	✓	partial

[Schedl et al., 2016]

Datasets: LFM-1b

[Schedl, ICMR, 2016]

- > 1B listening events
- Exact timestamps of listening events
- Demographic information of (anonymized) listeners
- Additional information describing the listeners' music preferences based on [Schedl and Hauger, SIGIR, 2015]
- Sample code to build a simple CF recommender
- Download: http://www.cp.jku.at/datasets/LFM-1b

- No audio files, nor content descriptors
- Selection of seed users for crawl might have biased results

LFM-1b: Data acquisition

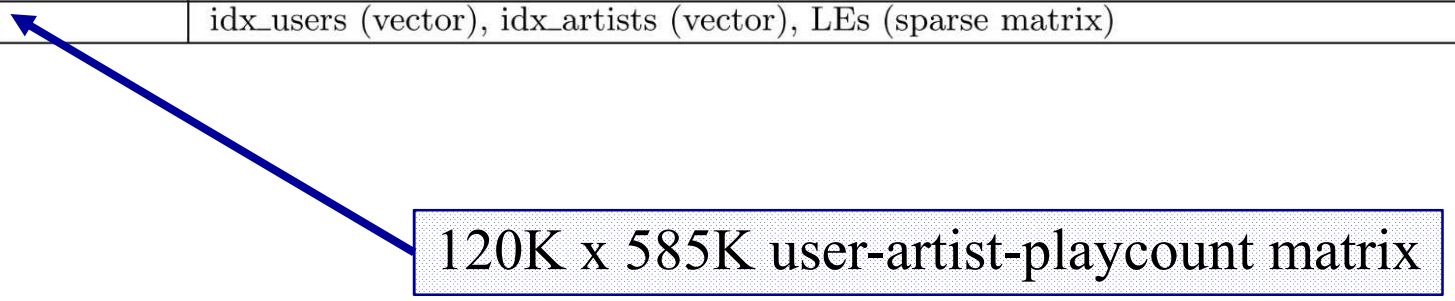
- Last.fm API calls (cf. http://www.last.fm/api)
- Seed list of 250 top tags
  → fetch top fans
  → 465K active users
  → random subset of 120K users
  → fetch their listening histories
- Listening events (LEs) fetched from January 2013 to August 2014
- LE = <user, artist, album, track, timestamp>

LFM-1b: Data acquisition and dataset content

- Data cleaning: remove users/artists with < 10 unique artists/users
- Available from http://www.cp.jku.at/datasets/LFM-1b
- Data as text files and HDF5/Matlab files
- Sample Python code for data import, simple statistical analysis and visualization, music recommendation experiments

LFM-1b: Dataset content

File	Content
LFM-1b_users.txt	user-id, country, age, gender, playcount, registered_timestamp
LFM-1b_users_additional.txt	user-id, novelty_artist_avg_month, novelty_artist_avg_6months, novelty_artist_avg_year, mainstreaminess_avg_month, mainstreaminess_avg_6months, mainstreaminess_avg_year, mainstreaminess_global, cnt_listeningevents, cnt_distinct_tracks, cnt_distinct_artists, cnt_listeningevents_per_week, relative_le_per_weekday1, ... relative_le_per_weekday7, relative_le_per_hour0, ... relative_le_per_hour23
LFM-1b_artists.txt	artist-id, artist-name
LFM-1b_albums.txt	album-id, album-name, artist-id
LFM-1b_tracks.txt	track-id, track-name, artist-id
LFM-1b_LEs.txt	user-id, artist-id, album-id, track-id, timestamp
LFM-1b_LEs.mat	idx_users (vector), idx_artists (vector), LEs (sparse matrix)



Statistical analysis: Items

Item	Number
Users	120,322
Artists	3,190,371
Albums	15,991,038
Tracks	32,291,134
Listening events	1,088,161,692
Unique <user, artist> pairs	61,534,450

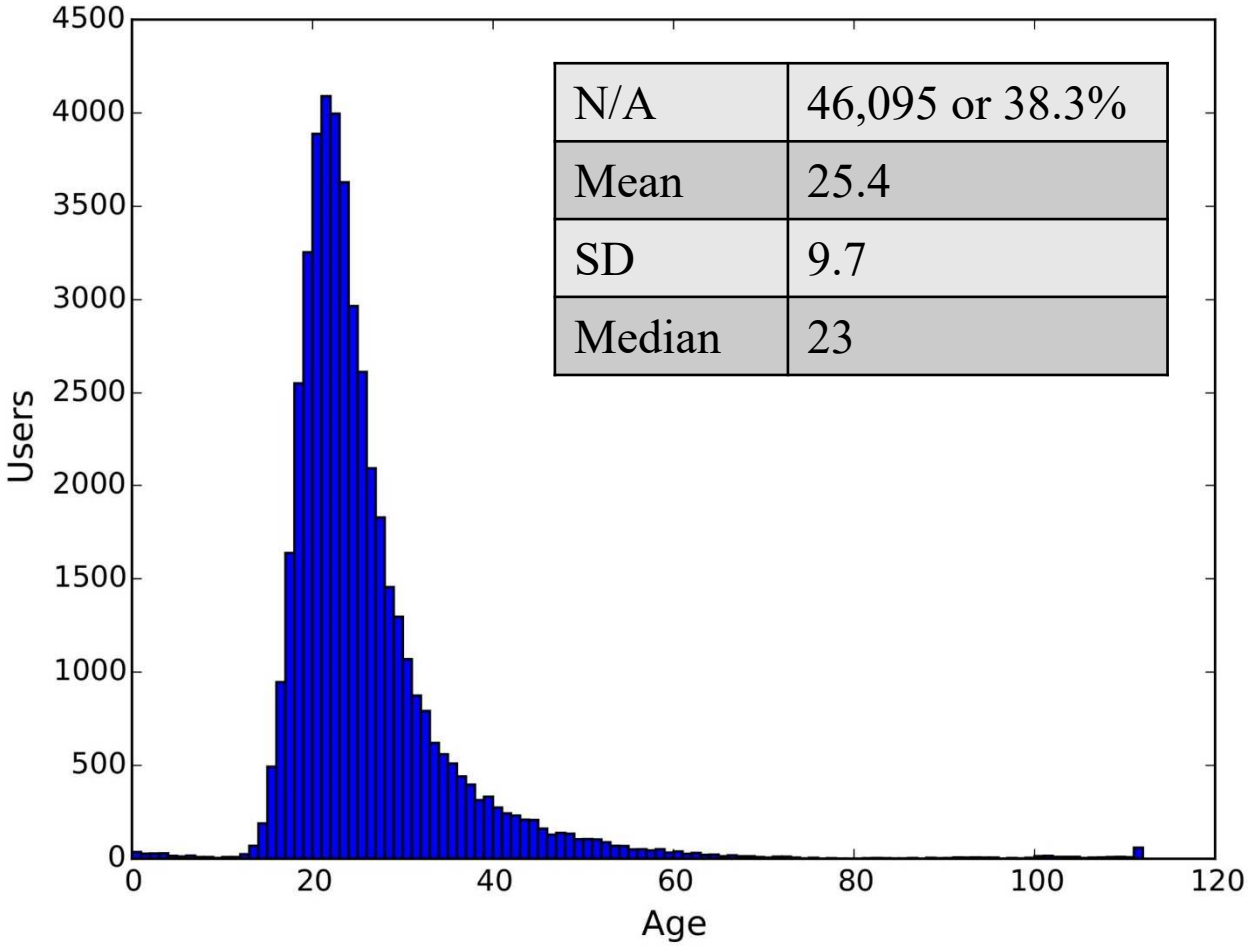
Statistical analysis: Country

Country	No. of users	Pct. in dataset
US	10255	18.581 %
RU	5024	9.103 %
DE	4578	8.295 %
UK	4534	8.215 %
PL	4408	7.987 %
BR	3886	7.041 %
FI	1409	2.553 %
NL	1375	2.491 %
ES	1243	2.252 %
SE	1231	2.230 %
UA	1143	2.071 %
CA	1077	1.951 %
FR	1055	1.912 %
N/A	65132	54.131 %

Statistical analysis: Gender

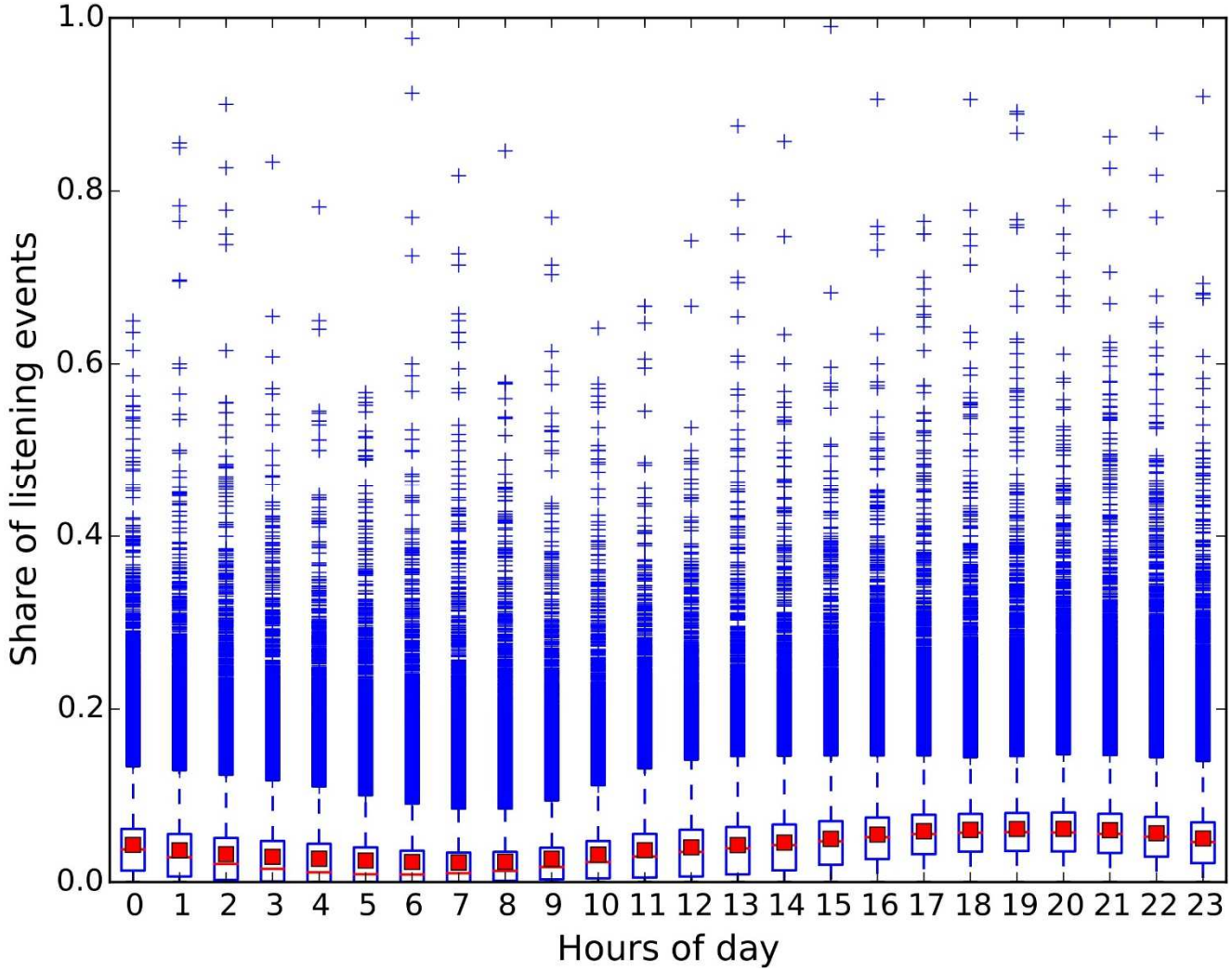
Gender	No. of users	Pct. in dataset
Male	39969	71.666 %
Female	15802	28.334 %
N/A	64551	53.649 %

Statistical analysis: Age

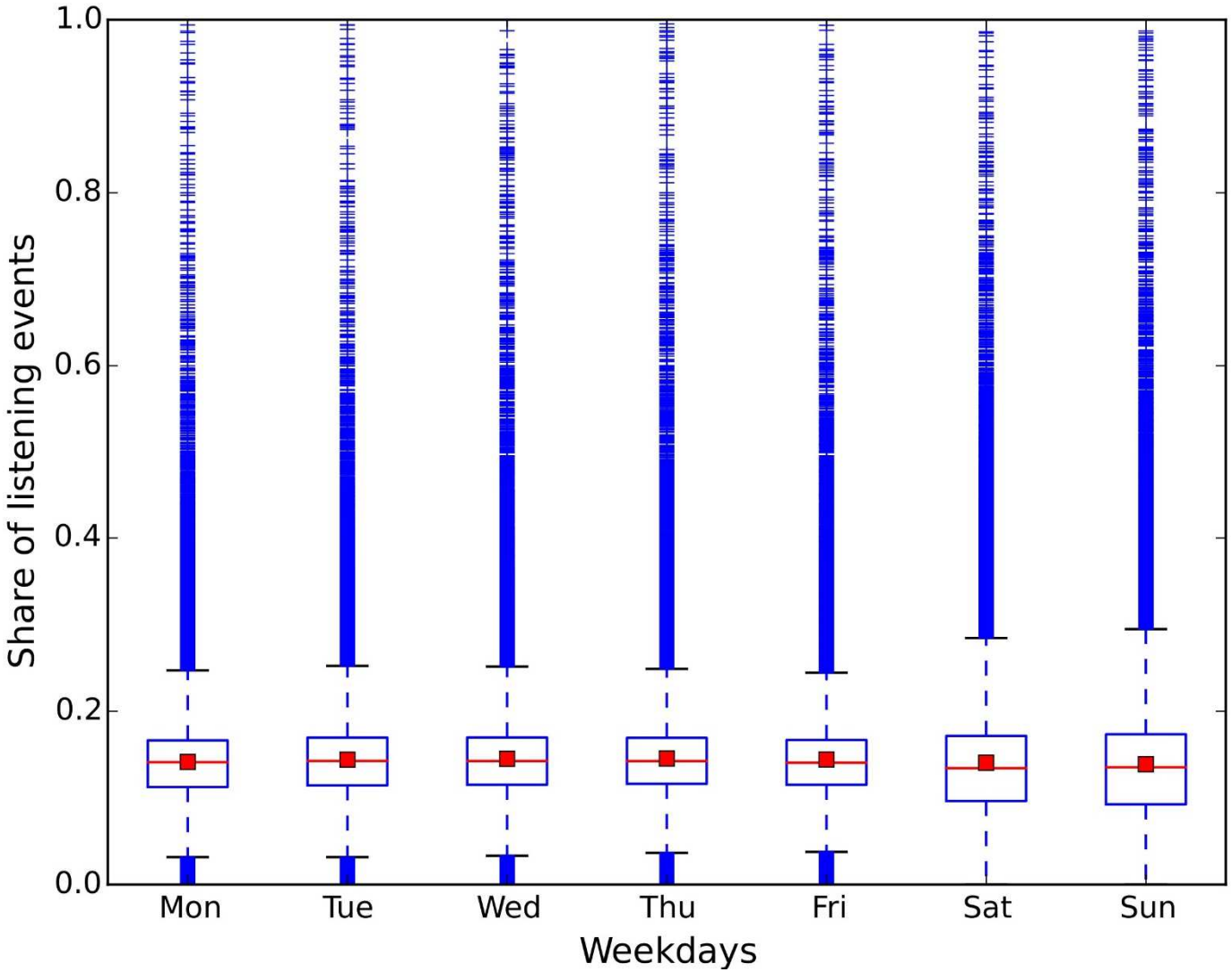


N/A	46,095 or 38.3%
Mean	25.4
SD	9.7
Median	23

Statistical analysis: Hour of day



Statistical analysis: Day of week



Statistical analysis: Novelty & Mainstreaminess

Novelty: share of new artists listed to for the first time, averaged over time windows of 12 months

Mainstreaminess: overlap between user's listening history and global listening history of all users

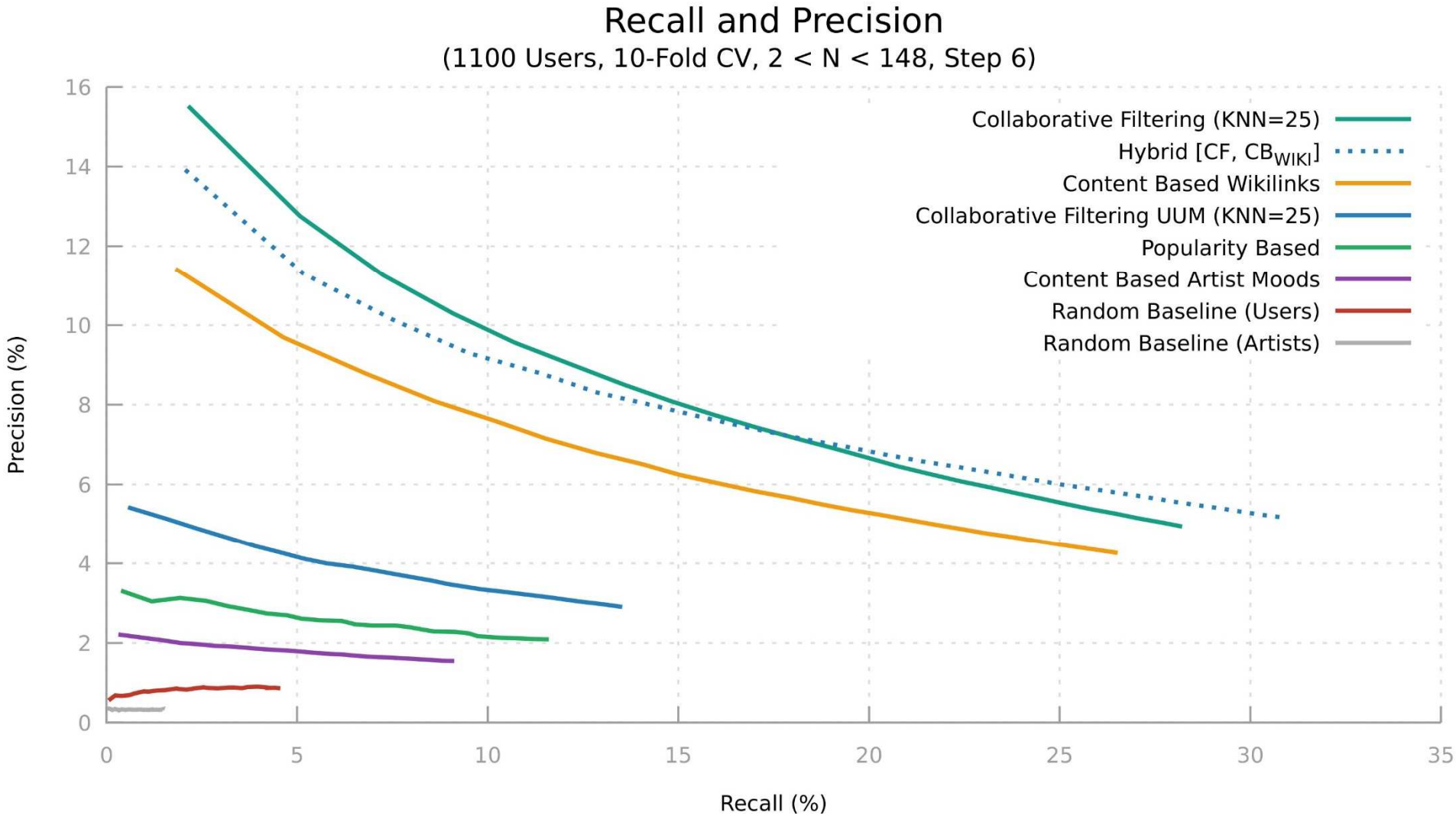
	Novelty	**Mainstreaminess**
Users tend to listen to a lot of new music and show a quite diverse consumption behavior.		
Mean	0.504	0.054
Std.	0.211	0.048

Use case: music recommender systems

PB	popularity-based recommender
CF	user-based collaborative filtering (memory-based)
CF-UUM	demographic filtering (based on similarity of age, gender, and country)
CB	content-based (artist similarity via **Wikipedia** links and **Allmusic** moods)
Hybrid	late fusion of normalized CF and CB artist ranking scores
RB	random baseline model: randomly picks **users** or **artists**

- Artist recommender system
- 10-fold CV on listening histories for each user
- Precision/recall for varying numbers *N* of recommender artists
- Code for simple CF recommender available from www.cp.jku.at/datasets/LFM-1b

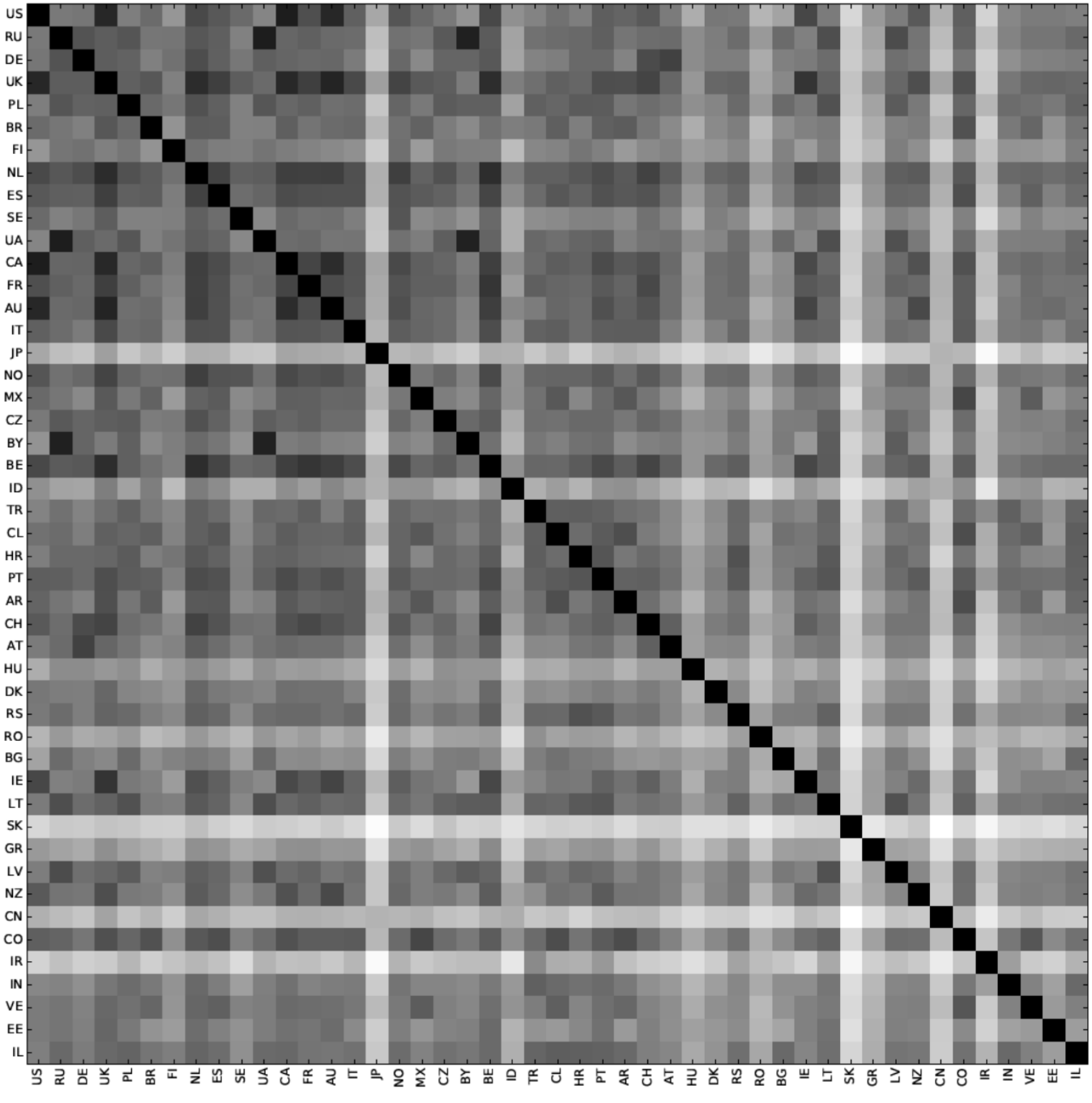
Use case: music recommender systems



Music preferences for some countries

U.S.A.		Japan		Finland	
Genre tag	PC	Genre tag	PC	Genre tag	PC
Rock	12.51	Rock	16.01	Rock	11.31
Alternative	9.63	Alternative	8.37	Metal	11.15
Alternative rock	5.86	J-pop	5.77	Alternative	7.30
Metal	4.77	Pop	4.56	Alternative rock	4.56
Pop	3.62	Metal	4.55	Hard rock	4.28
Indie	3.59	Alternative rock	4.26	Heavy metal	3.44
Hard rock	3.12	Indie	3.63	Death metal	2.74
Indie rock	3.09	Electronic	2.29	Classic rock	2.61
Classic rock	2.92	Hard rock	2.24	Pop	2.21
Electronic	2.33	Classic rock	2.23	Indie	2.13
Dance	2.21	Visual Kei	2.03	Electronic	2.00
Psychedelic	1.84	Indie rock	2.02	Indie rock	1.75
Blues	1.77	Heavy metal	1.68	Dance	1.71
Hip-Hop	1.72	Dance	1.66	Progressive rock	1.67
Punk	1.61	Punk	1.53	Nu metal	1.57
Heavy metal	1.49	Psychedelic	1.45	Progressive	1.50
Singer-songwriter	1.34	Anime	1.43	Power metal	1.46
Progressive	1.25	Electronica	1.43	Punk	1.45
Electronica	1.24	Blues	1.18	Alternative metal	1.32
Progressive rock	1.16	Japanese rock	1.17	Psychedelic	1.18
New Wave	1.08	Progressive rock	1.06	Hip-Hop	1.10
Punk rock	1.03	Pop punk	0.91	Electronica	0.90
Nu metal	0.99	Nu metal	0.86	Speed metal	0.89
Alternative metal	0.85	Progressive	0.86	Blues	0.84

Country similarities w.r.t. music taste



Evaluation

Evaluating What? Music Similarity?

- Three different genres?



- Which go together?

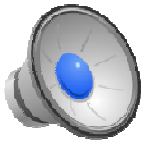






- Which are more similar?






Evaluating music recommender systems

- Recommendation can be seen as a special case of a **retrieval task**:
  - "query" is implicitly given (e.g., user's listening history)
  - retrieved items are music items (tracks, albums, artists)
  - analogous to retrieval, we have scores for each item → can build a ranked document/item list
  - full armory of performance measures used in retrieval is available

- Recommendation as a **classification task**:
  - predicting ratings for unknown items, based on known user rating
  - some additional evaluation strategies are possible

- Recommendation as a **user-centric task** aimed at satisfying the listener

Evaluation under retrieval aspects

- Compare retrieved/recommended music items and items truly listened to
- Recommended item is relevant if the user really listened to it
- Offline testing
- Automatic evaluation method

Performance Measures:

- Recall and Precision
- F-measure
- Precision a k documents (also Precision@k or P@k)
- Average Precision (AP)
- Mean Average Precision (MAP)

Recall and Precision

- Result to a seed item is an *unordered* set of documents.

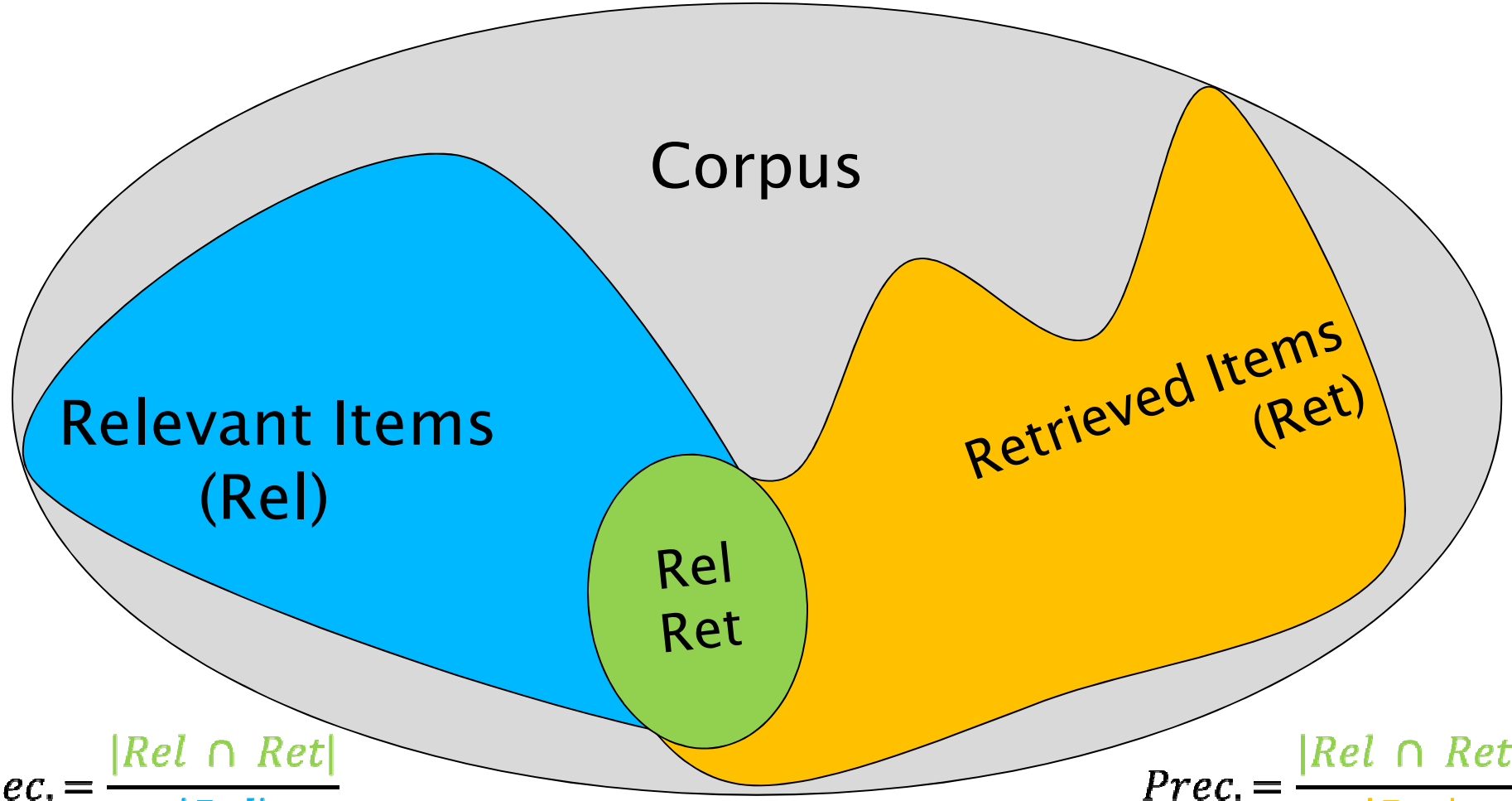
$$Recall = \frac{|Rel \cap Ret|}{|Rel|}$$

- Recall models how exhaustively the search results satisfy the user's information/entertainment need.

$$Precision = \frac{|Rel \cap Ret|}{|Ret|}$$

- Fraction of relevant items among recommended items.

Recall and Precision

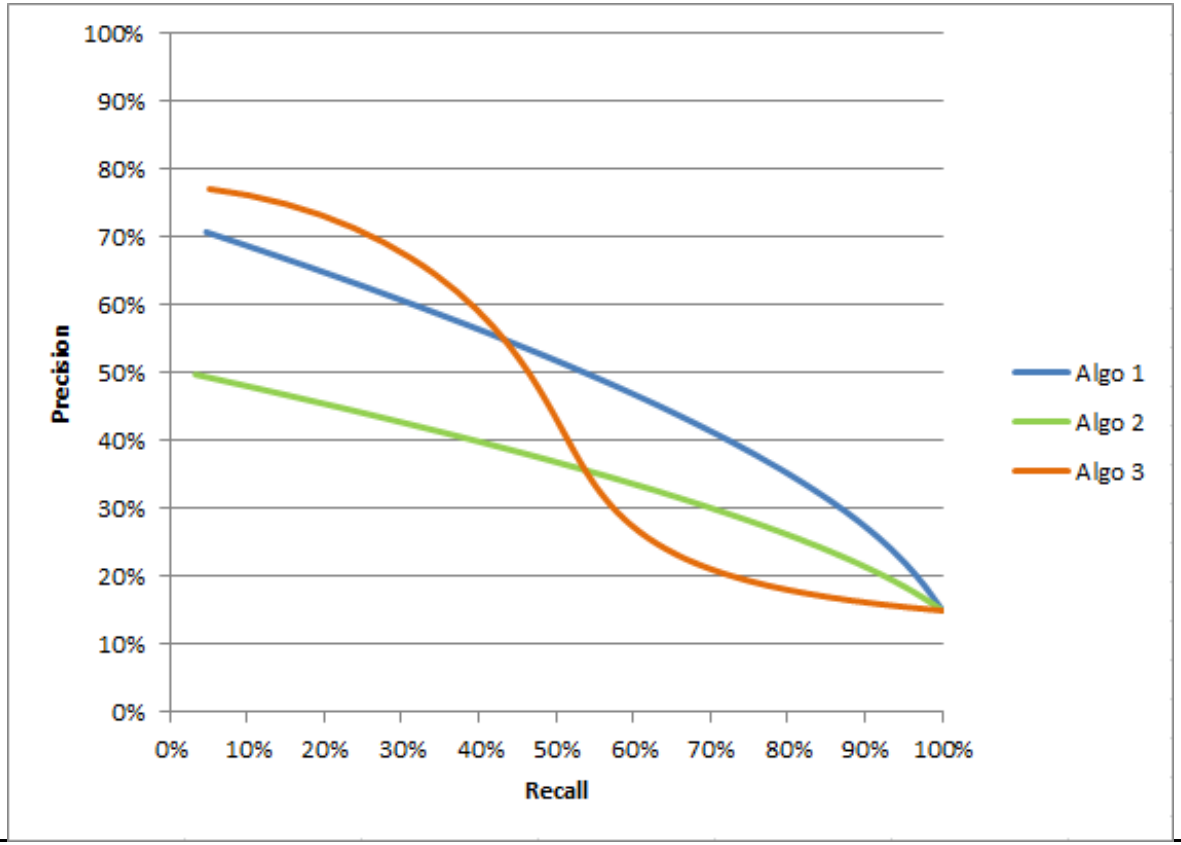


$$Rec. = \frac{|Rel \cap Ret|}{|Rel|}$$

$$Prec. = \frac{|Rel \cap Ret|}{|Ret|}$$

Recall and Precision

- Recall and precision varies, dependent on the number of retrieved items (usually, inverse relationship)
  → plots showing "precision at 11 standard recall levels"
  → requires parameter to vary



http://blog.cluster-text.com/tag/precision-and-recall/

F-measure

- Sometimes also referred to as *F_1 score* or *F-score*

- Harmonic mean of precision and recall:

$$F = 2 \times \frac{Precision \times Recall}{Precision + Recall} \qquad F@k = 2 \times \frac{Precision@k \times Recall@k}{Precision@k + Recall@k}$$

- Aggregate measure, taking into account both precision and recall
  → facilitates easy comparison between different algorithms

- Between the values of the recall and precision, usually closer to the smaller of the two
  → high *F-measures* are only possible if precision and recall high

Precision@k (P@k)

- Assumption: user is in general not interested in all items, but only looks at a number of *k* highly ranked items
- *P@k* assumes that user inspects the *k* items in an *arbitrary order*, and the user inspects *all of them*.

$$P@k = \frac{|Rel \cap Ret[1 \ldots k]|}{k}$$

Ret[1...k] is the top *k* items returned

Average Precision

- Problem of *P@k*: what should be taken as value of *k*? 10? 50? 100?
- Solution: a measure that combines precision values at all possible recall levels
- For every relevant item *d*, compute precision for the recommended items (result list) up to *d*

$$AP = \frac{1}{|Rel|} \times \sum_{i=1}^{|Ret|} relevant(i) \times P@i$$

relevant(i) = 1 iff the i^{th} retrieved item is relevant, 0 otherwise

- If a relevant item does not appear in *Ret*, its precision is 0.
- Implicitly models recall, because accounts for relevant items not in result list.

Mean Average Precision (MAP)

- So far, performance measures were defined on a single query/seed.
- In practice, when evaluating recommendation algorithms, we are interested in how well they perform for a variety of different music items (genres, artists, songs)

$$MAP = \frac{\sum_{i=1}^{|I|} AP(i)}{|I|}$$

I is the set of items, *AP(i)* is the average precision for query/item *i*

Evaluation under classification aspects

- Predict ratings for unknown data items
- Offline testing
- Automatic evaluation method

Performance Measures:

- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)
- (Rank Correlation)

Mean Absolute Error (MAE)

- RS predict **ratings** for unknown data items (e.g., on 5-point Likert scale)
- Measure how close predicted ratings are to true ratings

$$MAE = \frac{1}{|T|} \cdot \sum_{(u,i) \in T} |r_{u,i}' - r_{u,i}|$$

T … test set
u … user
i … item
$r_{u,i}'$ … predicted rating
$r_{u,i}$ true rating

- No distinction between high-ranked and low-ranked items

Root Mean Squared Error (RMSE)

- De-facto standard in evaluating rating-based RS
- In contrast to MAE, RMSE disproportionally *penalizes large prediction errors* (squared!)

$$RMSE = \sqrt{\frac{1}{|T|} \cdot \sum_{(u,i) \in T} \left(r'_{u,i} - r_{u,i}\right)^2}$$

T … test set
u … user
i … item
$r_{u,i}{}'$ … predicted rating
$r_{u,i}$ true rating

- Sometimes normalized to range of ratings ($r_{max} - r_{min}$); ranking remains the same

Rank Correlation

- Used when RS produces a **ranked list of items** that is shown to users
- Requires actual reference order as ground truth (e.g. sorting by ratings or song skipping events)
- Can be used as measure how similar two recommender systems behave

- Quantifies to which extent two rankings agree:
    - **Spearman's Rho ρ:** similar to Pearson's coefficient, but for ranks
    - **Kendall's Tau τ:** relates number of correctly ranked item pairs and incorrectly ranked item pairs

User-centric Evaluation

- Problem with all quantitative effectiveness measures so far:
  - Do they really assess if the recommended items satisfy the user?
  - What does "satisfy" mean?

- Aspects to consider:
  - ***Similarity*** (items should match the seed and be similar to each other)
  - ***Diversity*** (recommended items should not be too similar/boring)
  - ***Novelty / Familiarity*** (has the user already seen the item?);
    system can reach high accuracy just by making "easy" predictions (e.g., recommend always popular songs or songs by artists loved by user), but these are usually useless

User-centric Evaluation

- Aspects to consider:
  - ***Serendipity*** (user wants to discover something exciting, unexpected); e.g., interesting item from another genre that the user usually does not like); hard to measure (contrasting accuracy)
  - ***Explainability*** (recommender system should explain *why* an item was recommended); e.g., list similar users and their tastes
- **Need for user-centric evaluation, focusing on user satisfaction!**

User-centric Evaluation

- Asking real users is important to assess user satisfaction!

- Several strategies:
  - **Qualitative methods**
    surveys, structured interviews, …
    (explicitly ask users about their experiences with the RS)
  - **Quantitative methods**
    manual accuracy feedback for recommended items
    (ideally multifaceted, e.g., similarity, serendipity/discovery, suitability in current listening context, …)
  - **Implicit methods**
    observe user behavior, analyze logs

- Research: laboratory setting, artificial train/test set split, cross-fold validation
- Industry: A/B testing, in productive systems

Summary

- Various datasets for music recommender systems exist, with highly varying properties

- Well established evaluation metrics from IR, ML, and RS research available, but real user needs (e.g., serendipity or entertainment) often neglected