Recommender Systems Non-accuracy Metrics

Professor Robin Burke

Spring 2019

Non-accuracy metrics

- So far
  - (offline) Metrics compare what was recommended with the test data
- There might be other desirable properties of recommendation lists
  - Diversity
  - Novelty
- There might be other desirable properties of the recommendation algorithm generally
  - Catalog coverage
  - Fairness
- Not really covered in the book

Non-accuracy metrics

- Metrics that are not captured by comparison with test data
- But in many cases, users reported satisfaction (and click behavior) is correlated with non-accuracy metrics
- Two types
  - Local (list-wise) measures
    - Example: diversity
  - Global (whole test set) measures
    - Example: item coverage

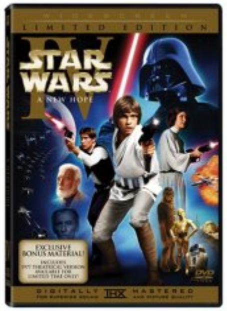
Non-accuracy metrics

- Local measures
  - Diversity
  - Novelty
  - Surprisal
  - Long-tail measures
- Global measures
  - Catalog coverage
  - Recommendation distribution
  - Long-tail coverage

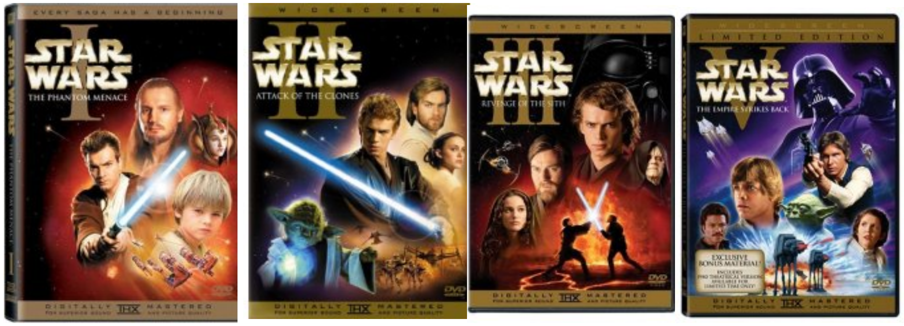
+

Diversity

- We want a recommender to provide a useful list
- Suppose I like this:



- Should I get this list?
  - not diverse



Some problems

- Risk
  - If the user doesn’t like one of these items
    - She won’t like any
- Utility
  - If the user already knows about one item
    - She probably knows about the others
  - Worst case
    - Copies of known items: news stories, etc.
- Choice confidence
  - If the items are too similar
    - The user might not believe that the search was “fair”

Local diversity

- The diversity of a given list
  - Whether the system as a whole recommends diverse items
    - Separate question
- Assume recommendation list
  - L = { i_0, … i_k }
- Assume similarity measure
  - Based on item features
  - sim(i,j) measures pairwise similarity
- Diversity
  - $d(L) = -\sum_{i,j \in L, i \neq j} sim(i,j)$

If diversity is the only objective for a recommender system, the best algorithm would be

- A. Content-based recommendation provided all the item features are known
- B. Collaborative recommendation provided the user base is diverse
- C. A recommender that chooses items randomly

Diversity-aware algorithm

- We can measure the extent to which items on the list are similar
  - prefer maximally diverse set for a given degree of predicted preference
- For example, ϵ greedy diversification re-ranking
  - Recommend k*10 items L
  - Put item_0 at position 0 in L'
  - For i = 1 to k
    - Let c = score(item_i)
    - From L, get items such that score >= c - ϵ
    - Put item at position i such that the diversity of L' is maximum
- Control the tradeoff between diversity and accuracy with ϵ

Intent-aware diversity

- An idea from information retrieval
  - Related to query ambiguity
- The user might have a specific intent, but an ambiguous query
  - "jaguar": car or animal?

- Intent-aware diversity =
  - Show some cars and some jungle cats
  - The user's possible intents are covered

Intent-aware recommendation

- Within the recommendation list
  - Identify dimensions of diversity (different intent)
  - For example, genre in movies
- Order the list so the best item in each category is at the top, before repeating categories
  - Example: best drama, best comedy, best action, 2nd best comedy, etc.

+

α-nDCG

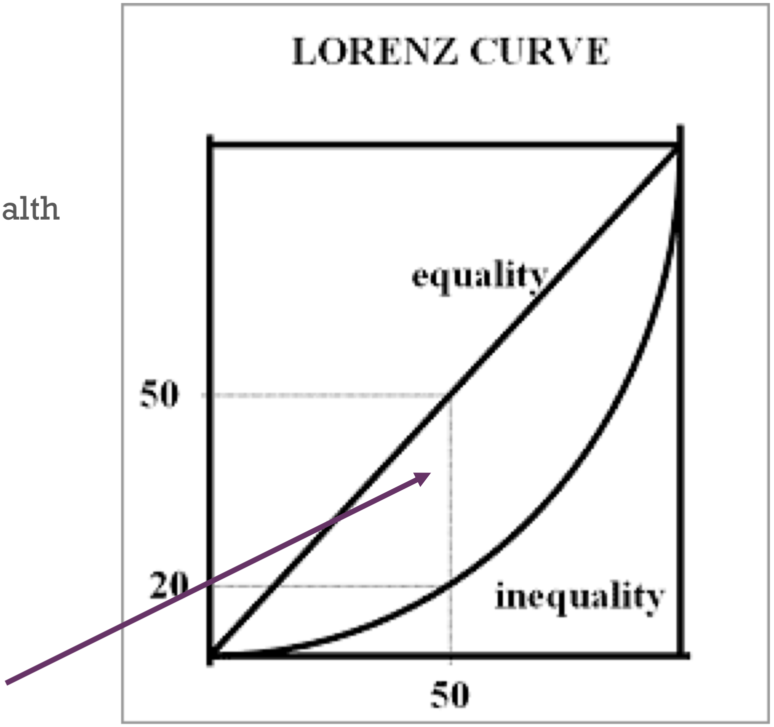
- Measure of intent-aware diversity
- Use regular NDCG definition
  - $DCG@10 = \sum_{l=1}^{10} \frac{2^{u_l} - 1}{log_2(i+1)}$
- But instead of the gain being binary
  - r(t, k-1) = number of items that have property t in list up to k-1
  - J(i_k, t) = 1 if retrieved item i_k has property t
  - $G[k] = \sum_{t=1}^{s} J(ik, t)(1 - \alpha)^{r(t,k-1)}$
- The idea
  - Use alpha to discount the gain achieved by redundant items

Novelty / Surprisal

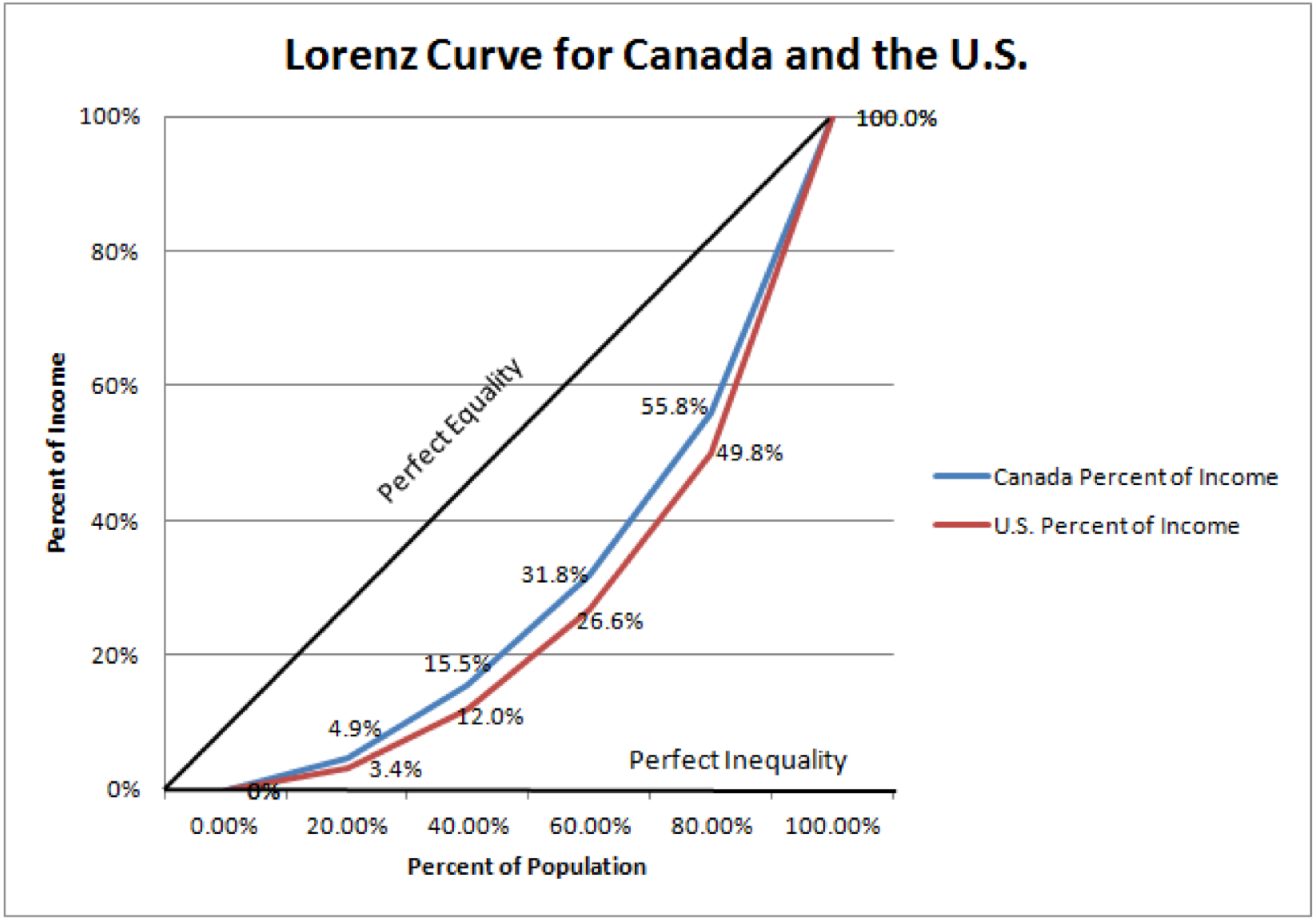
- Imagine a ”most popular” recommender
  - Same top items to everybody
- No novelty
  - Top items are likely to be well known
  - Not surprising
- Also not profitable
  - Recommend bananas, milk to grocery shoppers
  - They were going to buy those anyway

Gini index

- Measures inequality
  - Not Gini impurity
- Lorenz curve
  - cumulative proportion of population vs cumulative wealth
- In recommendation contexts
  - Wealth = # of recommendations
- 45 degree line means that x% of the items
  - Have x% of the recommendations
  - Equality
- Gini index
  - The area in between the equality line and the actual distribution



+ Incidentally



Gini index

- Look at the distribution of items in all the recommendation lists that you generate
- Measures how skewed the recommendations are
  - Do some items get recommended way more than others?
    - Maybe they are just better!
- But if two algorithms have the same accuracy
  - And one has a lower Gini index
  - You would probably prefer that one
    - Because it doesn’t have as much popularity bias

You measure some properties of your music recommender system and discover that the recommendation lists have high diversity but low novelty. You conclude

- A. That there must be a bug in your metric implementations because that isn't possible
- B. That the recommendation lists combine popular songs across a range of different musical styles
- C. That the recommendation lists are all of similar style but come from different places in the popularity distribution

Surprisal

- Probabilistic measure of the same idea

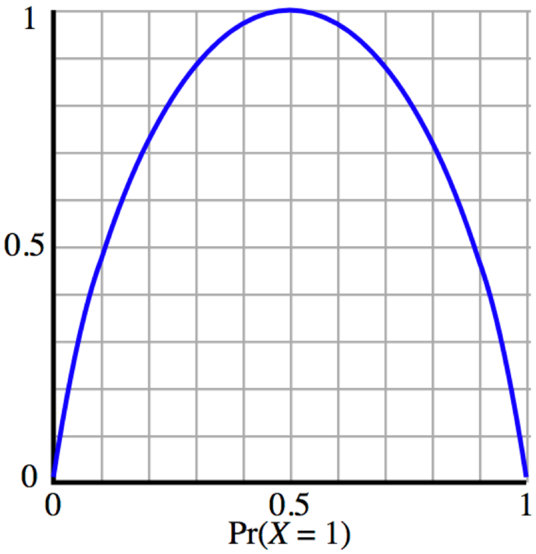
- Entropy is a measure of disorder in a system
  - if all outcomes are equally likely
  - entropy is maximum

- Quantified
  - -p(x) log p(x)

- In a recommendation context
  - p(i) refers to the frequency of <u,i> pairs in the training data

- If an item has high probability (close to 1)
  - It has lower entropy
    - -0.8 log (0.8) = .258
  - Low probability, also lower entropy
    - -0.1 log (0.1) = .332
  - Highest entropy (1) at 0.5

- Can sum over the whole recommendation list

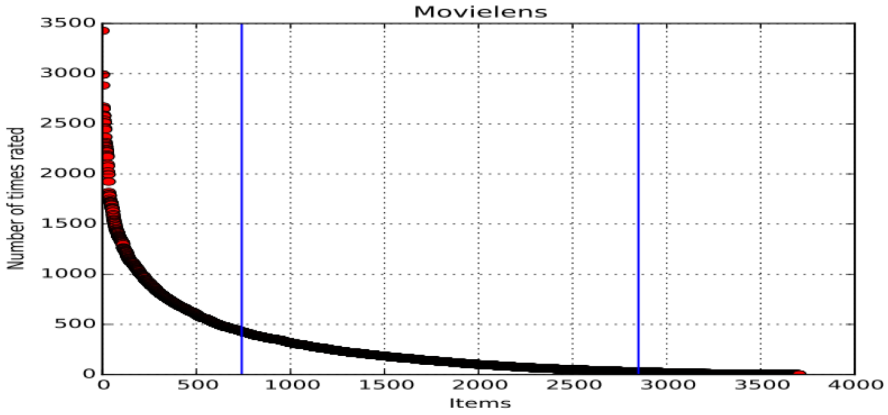


Surprisal

- Can measure the average surprisal of each recommendation list
- Again, maximally surprising list is easy
  - Pick items that evenly divide users
  - Not necessary the best recommendations
- Can we maximize surprisal with minimal accuracy loss?

Long-tail items

- Novelty and surprisal are measures, in part, how well the recommender does at recommending long-tail / cold start items



Other long-tail measures

- Percentage of long-tail items in the average list
  - APT
- Average list popularity
  - Average # of ratings for all items on a list
  - Lower means more long-tail items

In our research, we have found this is not a good measure

- But might indicate a concentration on a small number of these items
  - Coverage is a different measure
  - Need to look at more than just one list / user

Global measures

- Suppose I have 2 long-tail items A and B
- I put A and B in every recommendation list
- I will have 20% APT
    - Many recommendation algorithms have something like 1%
- But this is kind of unsatisfying
    - Lots of long-tail items aren’t being shown at all
- To capture this we need global measures
    - Look across all the recommendations being made

+

Personalization

- How different (personalized) are recommendations?
  - Do users get different lists?
- Why do I care about this?
  - Indicates to what extent the recommender system is capturing individual differences
  - If many users are similar, lower personalization methods might still have high accuracy

Personalization

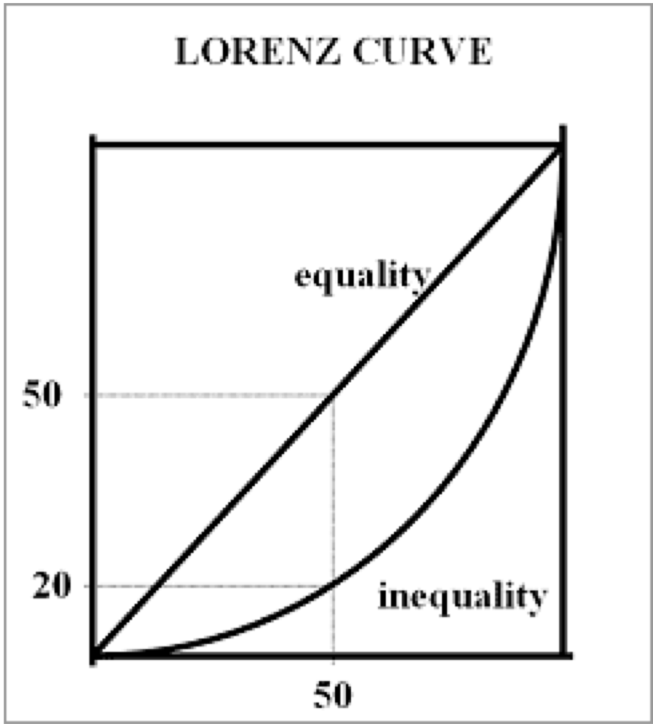
- Generate recommendation lists of size n for all users
  - compare pairwise overlap between lists
    - Jaccard coefficient
  - average of all pairs of lists
  - divide by n
- High value of this metric means that different items are appearing in the lists

You compare the recommendations produced by two algorithms and find that they have different accuracies, but both have low surprisal and low personalization. A possible explanation for this is:

- A: One of the algorithms is content-based and the other is collaborative
- B: One of the algorithms is favoring long-tail items more than the other
- C: The users of this system aren’t that different from each other

Gini index

- What is the distribution of recommendations?
- Do the bottom 50% of the items get 50% of the recommendations
- Same principle as looking at item popularity
  - Look at popularity in the recommended set



Catalog Coverage

- What percentage of items are recommended?
  - Like Gini index, but less nuance
  - Are there items that are never recommended?
    - Usually there are many
- Easy (and bad) to maximize
  - Maybe some items should never be recommended
  - Low catalog coverage is a problem
    - Similar to high Gini index

Non-accuracy metrics

- A large set of possibilities
- Generally in conflict with prediction accuracy
  - At least as measured in offline settings
  - A/B tests give quite different answers
- Because of this conflict, we have a multi-objective optimization problem
  - Must decide how much to value different aspects
- Users may differ in how much they value these aspects of recommendation
  - Adds potential complexity
  - Although non-diverse lists are generally considered bad

Practical importance

- Users actually care about these aspects of recommendation
- Requires a change of emphasis
  - Recommendation is about decision support
  - Not just rating prediction
- Open research area
  - Which metrics matter?
  - In which domains?
  - What is the right trade-off between metrics?
    - How is that influenced by domain characteristics?
- Hard to evaluate off-line
  - If your original algorithm didn't give diverse / novel results
    - You won't know what else the user would have liked

Another non-accuracy question

- Is your recommender system fair?
- Could mean a lot of different things.
- Topic for Thursday
  - See posted reading – it's short!