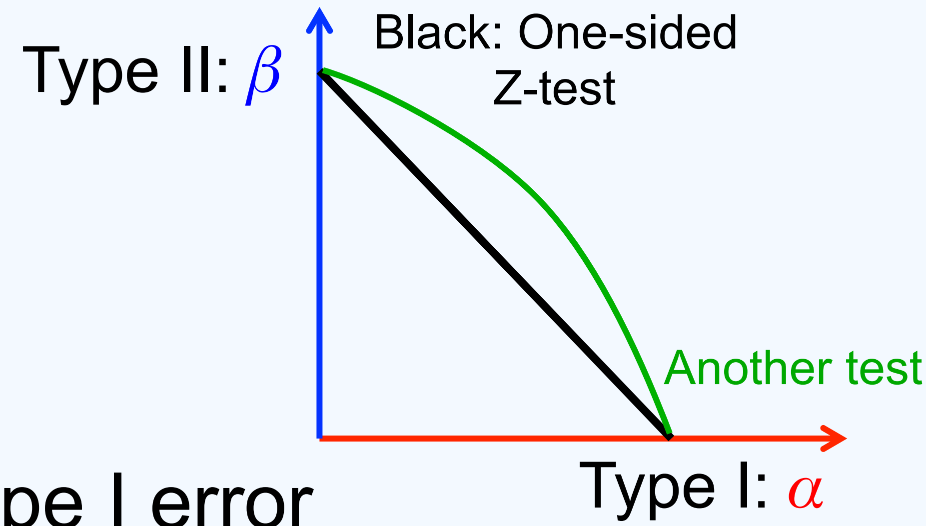
Announcements

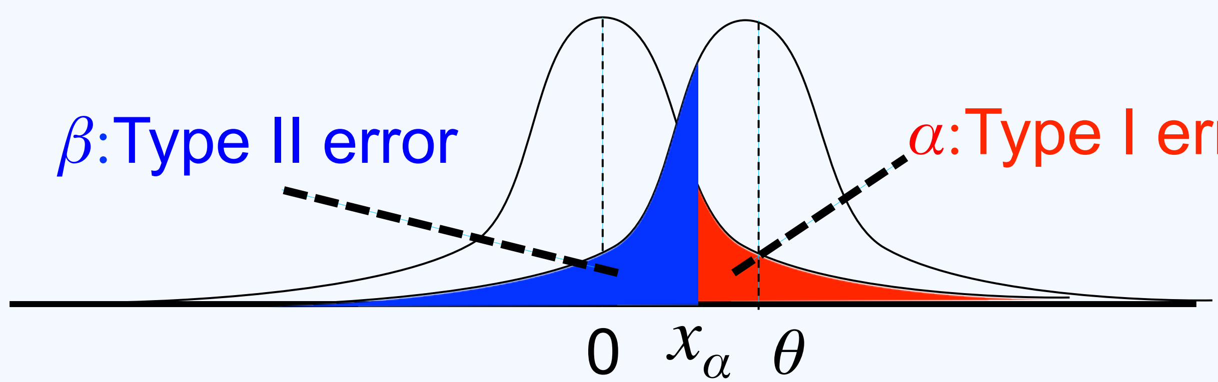
- Paper presentation
  - meet with me ASAP
  - 1st time: tell me what you will discuss
  - 2nd time: show me the slides
  - prepare for a few reading questions
- Project
  - meet with me ASAP
  - think about a problem that may use social choice, game theory, or mechanism design

Last time



- One-sided Z-test
  - we can freely control Type I error
  - for Type II, fix some $\theta \in H_1$

		Output	
		Retain	Reject
Ground truth in	H_0	size: $1-\alpha$	Type I: α
	H_1	Type II: β	power: $1-\beta$



How to do test for your problem?

- Step 1: look for a type of test that fits your problem (from e.g. wiki)
- Step 2: choose H_0 and H_1
- Step 3: choose level of significance α
- Step 4: run the test

Today: recommender systems

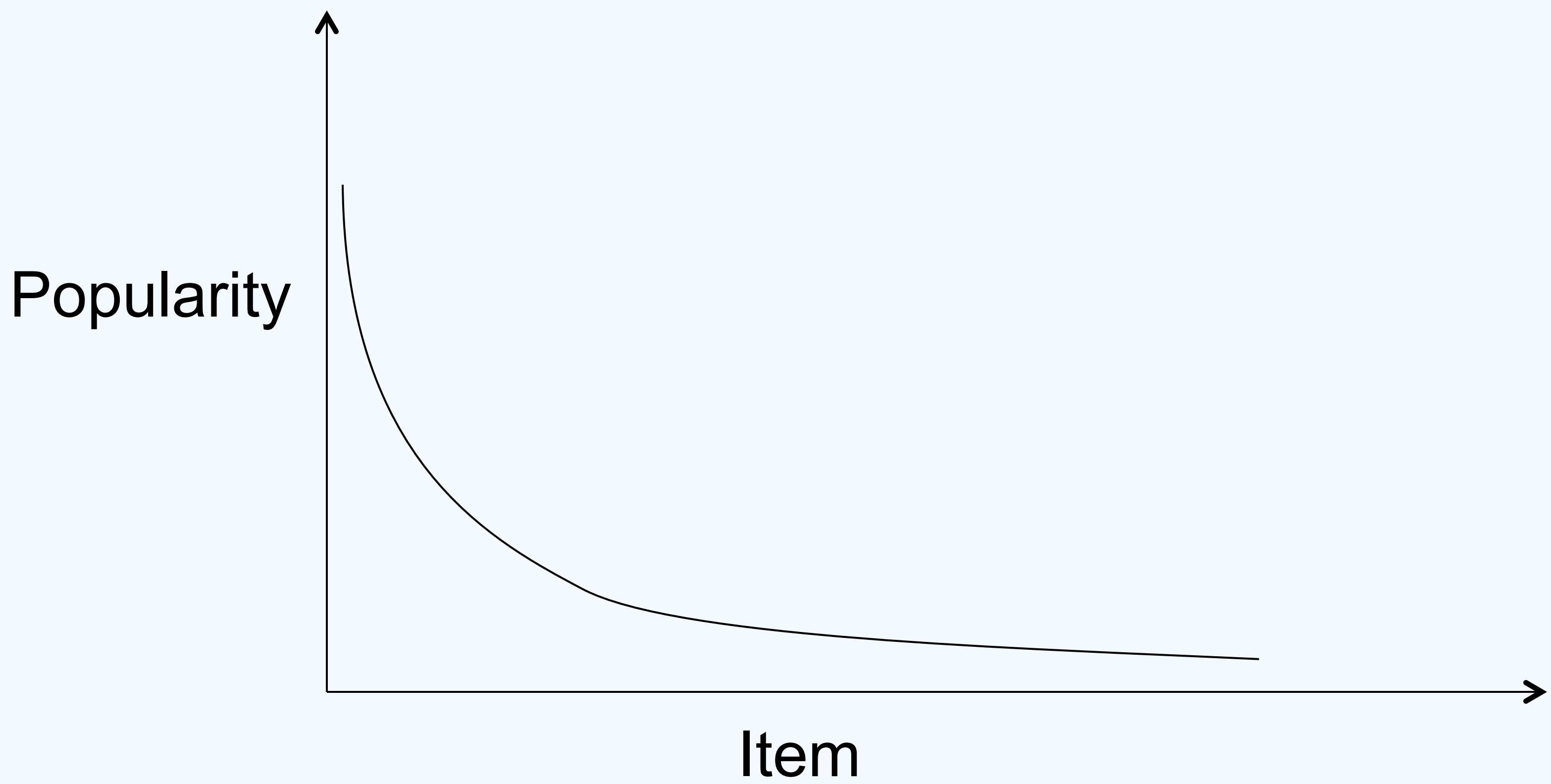
- Content-based approaches
  - based on user's past ratings on similar items computed using features
- Collaborative filtering
  - user-based: find similar users
  - item-based: find similar items (based on all users' ratings)

Applications

The Netflix challenge

- $1M award to the first team who can outperform their own recommender system CinMatch by 10%
- A big dataset
  - half million users
  - 17000 movies
  - a secret test set
- Won by a hybrid approach in 2009
  - a few minutes later another hybrid approach also achieved the goal

Exploring the tail



- Personalize to sell the “tail” items

The problem

- Given
  - features of users i
  - features of items j
  - users' ratings $r_i(j)$ over items
- Predict
  - a user's preference over items she has not tried
    - by e.g., predicting a user's rating of new item
- Not a social choice problem, but has a information/preference aggregation component

Classical approaches

- Content-based approaches
- Collaborative filtering
  - user-based: find similar users
  - item-based: find similar items (based on all users' ratings)
- Hybrid approaches

Framework for content-based approaches

- Inputs: profiles for items
  - K features of item j
    - $w_j = (w_{j1}, \ldots, w_{jK})$
    - $w_{jk} \in [0,1]$: degree the item has the feature
  - the user's past ratings for items 1 through j-1
- Similarity heuristics
  - compute the user's profile: $v_i = (v_{i1}, \ldots, v_{iK})$, $v_{ik} \in [0,1]$
  - recommend items based on the similarity of the user's profile and profiles of the items
- Probabilistic approaches
  - use machine learning techniques to predict user's preferences over new items

Example

	Animation	Adventure	Family	Comedy	Disney	Bluesky	rate
	1	1	1	0	0	1	?
	1	1	0	1	0	1	9
	1	0	1	1	1	0	8
	1	1	1	0	1	0	7
$v =$	0.8	0.8	0.75	0.85	0.75	0.9	

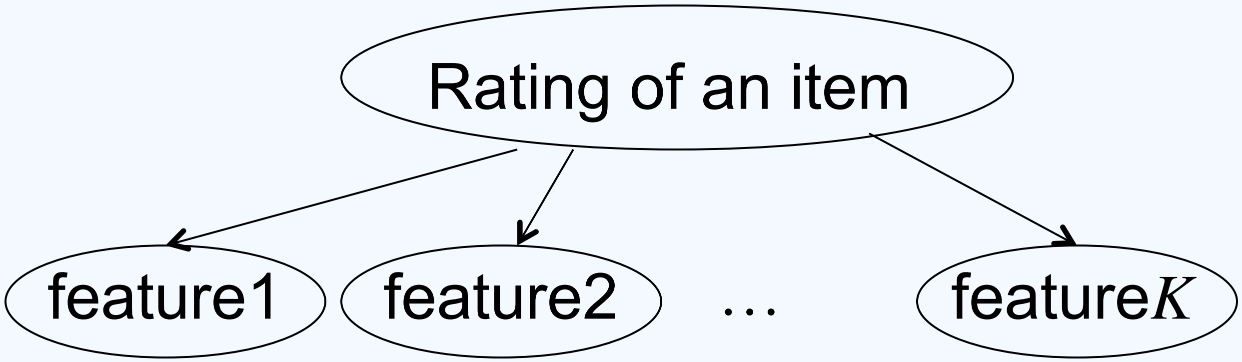
Similarity heuristics

- A possible way to define v_i
  - v_{ik} is the average normalized score of the user over items with feature k
- A possible way to define similarly measure
  - cosine similarity measure

$$\cos(v_i, w_j) = \frac{v_i \cdot w_j}{\| v_i \|_2 \| w_j \|_2} = \frac{\sum_{k=1}^{K} v_{ik} \cdot w_{jk}}{\sqrt{\sum_{k=1}^{K} v_{ik}^2} \sqrt{\sum_{k=1}^{K} w_{ik}^2}}$$

  - in the previous example, the measure is 0.68

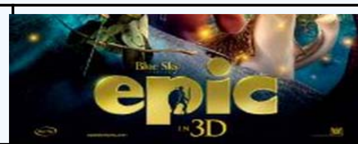
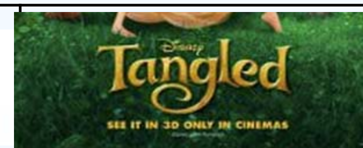
Probabilistic classifier



- Naïve Bayes model: suppose we know
  - $\Pr(r)$
  - $\Pr(f_k|r)$ for every r and k
  - learned from previous ratings using MLE
- Given $w_j = (w_{j1}, \ldots, w_{jK})$
  - $\Pr(r|w_j) \propto \Pr(w_j|r)\Pr(r) = \Pr(r)\,\Pi \Pr(w_{jk}|r)$
  - Choose r that maximizes $\Pr(r|w_j)$

Framework for collaborative filtering approaches

- Inputs: a matrix M.
  - $M_{i,j}$: user i's rating for item j

				
Alice	8	6	4	9
Bob	$\varnothing$	8	10	10
Carol	4	4	8	$\varnothing$
David	6	$\varnothing$	10	5

- Collaborative filters
  - User-based: use similar users' rating to predict
  - Item-based: use similar items' rating to predict

User-based approaches (1)

- Step 1. Define a similarity measure between users based on co-rated items
  - Pearson correlation coefficient between i and i^*
  - G_{i,i^*}: the set of all items that both i and i^* have rated
  - $\overline{M}_i$: the average rate of user i

$$sim(i,i^*) = \frac{\sum_{j \in G_{i,i^*}} (M_{ij} - \overline{M}_i) \cdot (M_{i^*j} - \overline{M}_{i^*})}{\sqrt{\sum_{j \in G_{i,i^*}} (M_{ij} - \overline{M}_i)^2} \sqrt{\sum_{j \in G_{i,i^*}} M_{i^*j} - \overline{M}_{i^*})^2}}$$

User-based approaches (2)

- Step 2. Find all users $i*$ within a given threshold
  - let N_i denote all such users
  - let N_i^j denote the subset of N_i who have rated item j

User-based approaches (3)

- Step 3. Predict *i*'s rating on *j* by aggregating similar users' rating on *j*

$$\hat{r}_i(j) = \frac{1}{|N_i^j|} \sum_{i^* \in N_i^j} r_{i^*}(j)$$

$$\hat{r}_i(j) = \frac{\sum_{i^* \in N_i^j} sim(i,i^*) r_{i^*}(j)}{\sum_{i^* \in N_i^j} sim(i,i^*)}$$

$$\hat{r}_i(j) = \overline{M}_i + \frac{\sum_{i^* \in N_i^j} sim(i,i^*)(r_{i^*}(j) - \overline{M}_{i^*})}{\sum_{i^* \in N_i^j} sim(i,i^*)}$$

Item-based approaches

- Transpose the matrix M
- Perform a user-based approach on M^T

Hybrid approaches

- Combining recommenders
  - e.g. content-based + user-based + item-based
  - social choice!
- Considering features when computing similarity measures
- Adding features to probabilistic models

Challenges

- New user
- New item
- Knowledge acquisition
  - discussion paper: preference elicitation
- Computation: challenging when the number of features and the number of users are extremely large
  - M is usually very sparse
  - dimension reduction

Recap: recommender systems

- Task: personalize to sell the tail items
- Content-based approaches
  - based on user's past ratings on similar items computed using features
- Collaborative filtering
  - user-based: find similar users
  - item-based: find similar items (based on all users' ratings)
- Hybrid approaches