2-4: Scoring and Ranking

Introduction to Recommender Systems

Introduction

- Last 2 lectures:
  - how to collect data
  - what we present to users
- This lecture: how to do it
  - what predictions to show
  - how to rank

Learning Objectives

- Understand several ways of computing and displaying predictions
- Understand how to rank items with sparse, time-shifting data
- Understand several points in the design space for prediction and recommendation, and some of their tradeoffs

Overview

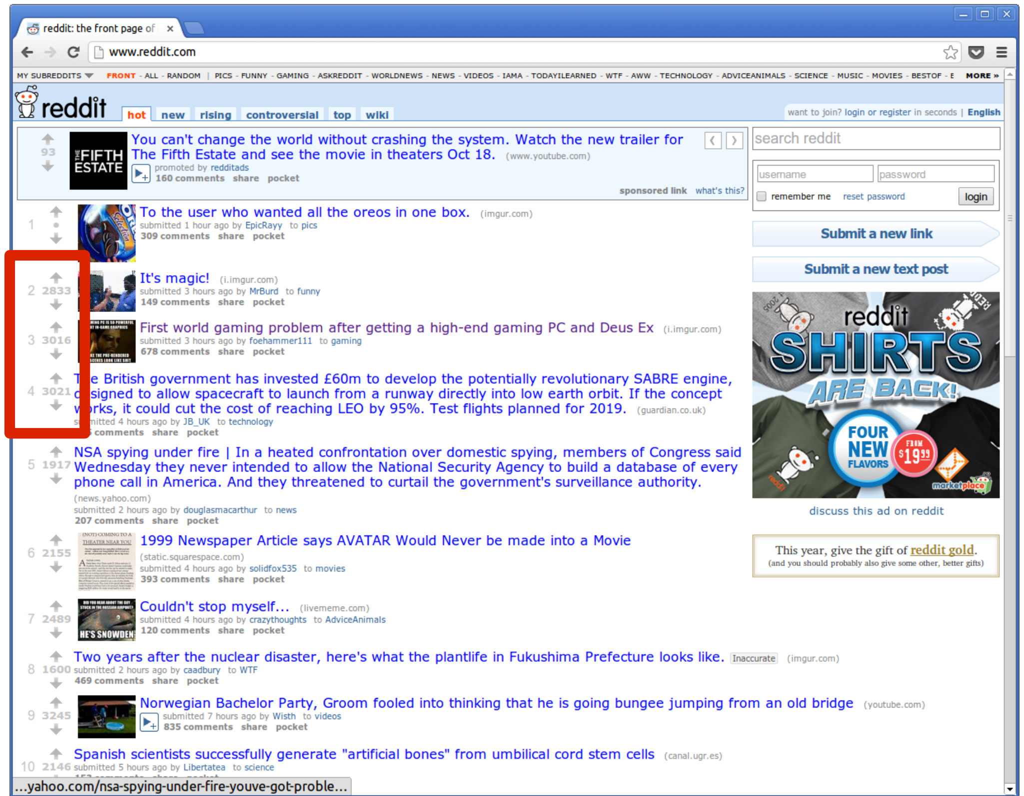
- Example
- Displaying Aggregate Preferences (*predict*)
- Ranking Items (*recommend*)

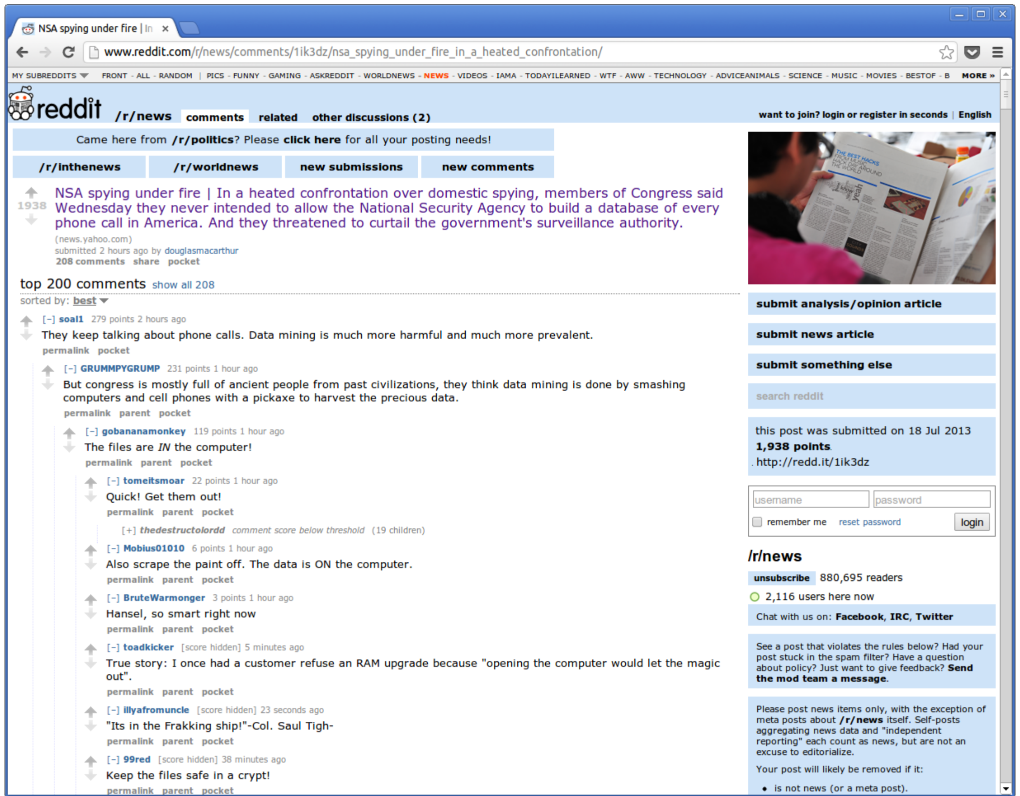
Overview

- **Example**
- Displaying Aggregate Preferences (*predict*)
- Ranking Items (*recommend*)

Example - Reddit

- Social news aggregator
- Non-personalized news recommender
- Users vote on items to determine top item





Overview

- Example
- **Displaying Aggregate Preferences (*predict*)**
- Ranking Items (*recommend*)

Simple Display Approaches

- Average rating / upvote proportion
- Net upvotes / # of likes
- % >= 4 stars (‘positive’)
- Full distribution

Goal of Display

To help users decide to buy/read/view the item.

Simple Display Approaches (again)

- Average rating / upvote proportion
  - Of people who vote, do they like it?
  - Doesn't show popularity
- Net upvotes / # of likes
  - Shows popularity
  - No controversy
- % >= 4 stars (‘positive’)
- Full distribution
  - Complicated

Example: Amazon.com

Reddit

Overview

- Example
- Displaying Aggregate Preferences (*predict*)
- **Ranking Items (*recommend*)**

Ranking

- What do you put at the top of Reddit?
- What is at the top of the e-Bay search list?
- You don't have to rank by prediction

Why not rank by score?

- Too little data (one 5-star rating)
- Score may be multivariate (histogram)
- Domain or business considerations
  - Item is old
  - Item is ‘unfavored’

Ranking Considerations

- Confidence
  - How confident are we that this item is good?
- Risk tolerance
  - High-risk, high-reward
  - Conservative recommendation
- Domain and business considerations
  - Age
  - System goals

Damped means

- Problem: low confidence w/ few ratings
- Solution: assume that, without evidence, everything is average
- Ratings are evidence of non-averageness
- k controls strength of evidence required

$$\frac{\sum_u r_{ui} + k\mu}{n + k}$$

Confidence Intervals

- From the reading: lower bound of statistical confidence interval (95%)
- Choice of bound affects risk/confidence
  - Lower bound is conservative: be sure it's good
  - Upper bound is risky: there's a chance of amazing
- Reddit uses Wilson interval (for binomial) to rank comments

Domain Consideration: Time

- Reddit: old stories aren't interesting
  - even if they have many upvotes!
- eBay: items have short lifetimes

Scoring news stories

- Hacker News

$$\frac{(U - D - 1)^{\alpha}}{(t_{\mathrm{now}} - t_{\mathrm{post}})^{\gamma}} \times P$$

- Net upvotes, polynomially decayed by age
- Old items scored mostly by vote
- Multiplied by item penalty terms
  - incorporate community goals into score

Reddit algorithm (c. 2010)

$$\log_{10} \max\left(1, |U - D|\right) + \frac{\operatorname{sign}(U - D)\, t_{\mathrm{post}}}{45000}$$

- Log term applied to votes
  - decrease marginal value of later votes
- Time is seconds since Reddit epoch
- Buries items with negative votes
- Time vs. vote impact independent of age
- Scores news items, not comments

Ranking Wrap-Up

- There are some theoretically grounded approaches (confidence interval, damping)
- Many sites use ad-hoc methods
- Most formulas have constants, will be highly service-dependent
- Can manipulate for ‘good’ or ‘evil’
- Build based on domain properties, goals

Predict with sophisticated score?

- Theoretically a fine thing to do
- Be careful with transparency/scrutability
  - If you say 'average rating' for damped mean, and show ratings, users may be confused
  - Most important case (low ratings) also easiest to hand-verify

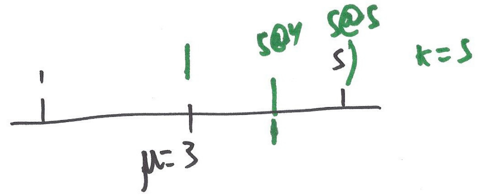
Conclusion

- Sparsity, inconsistency, temporal concerns make data messy
- Simple scoring doesn't necessarily match the domain or business
- There are good ways to deal with this (decay, time, penalties, damping)
- We'll see more normalizations later

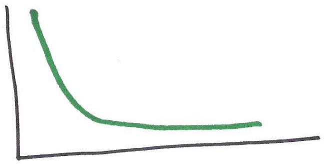
2-4: Scoring, Ranking, and Normalization

$$\frac{\sum_u r_{ui} + k\mu}{n + k}$$

strength required



$$\frac{(\overbrace{U-D-1}^{\text{net upvotes}})^{\alpha=0.8}}{(\underbrace{t_{now}-t_{post}}_{\text{age}})^{\gamma=1.8}} \cdot \underline{P}$$



$$\log_{10} \max(1, \underline{|U-D|}) + \frac{\text{sign}(U-D)\, t_{post}}{45,000}$$

pred.

ratings

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

user a item i

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

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

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

similarity

predictive-ness

$$w_{a,u} = \frac{\sum_{i=1}^{m} (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sigma_a \sigma_u}$$

- m small?
- unary?

top-n?