Crawling in the books

Daniel Šmít

Initial data manipulation

- set up locale, clean missing and malformed data
- remove single ratings (user with 1 book, book with 1 rating)
- bundle ratings with age and geolocation of user
- filter LOTR books, or all Tolkien books (latter better for validation)

Insights

- plot ratings by age, geolocation, user ratings
- compute rating-count, and mean rating for LOTR books

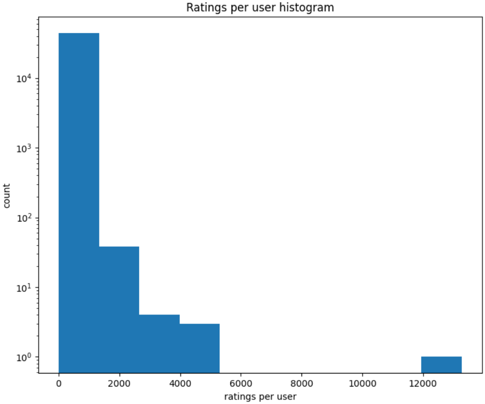

Figure: User rating count histogram

General descriptors

- general ratings statistics by book (count, mean rating, rating std)
- LOTR set books statistics (count, mean and std of rating)
- filtering by low count or low ratings

Rating group processing

Prepare user-based groups:

- select user-based rating groups $\{\gamma_i\}$ which contain at least one LOTR book
- *count LOTR books* $\{\lambda_i\}$ and *count all books* $\{\beta_i\}$ in each user group i

Prepare metrics for each relevant books:

1. book belongs to $J > 5$ rating groups $\{\gamma_1, \ldots, \gamma_J\}$
2. mean number of LOTR books co-ocurring in rating groups $\Lambda = \frac{1}{J}\sum_{j=1}^{J} \lambda_j$
3. mean size of rating groups $B = \frac{1}{J}\sum_{j=1}^{J} \beta_j$
4. normalized weight as $W = \frac{\Lambda}{B}$

Normalization gives high weight to groups where a book is accompanied by a large proportion of LOTR/Tolkien books.
Note that we could have chosen another metric for normalized weight, possibly involving mean rating, or rating std.

Ranking

Ranking of Tolkien co-rated books:

- compute weighted count $\kappa = W \cdot J$ of each book in $\{\gamma_1, \dots, \gamma_J\}$
- rank book within Tolkien co-rated set by κ
- rank all general books by their count κ' and filter Tolkien co-rated
- compute rank-gain for each book in Tolkien co-rated set $\kappa - \kappa'$

The rank gain represents how much more frequently a book co-occurs with Tolkien books relatively to other book as compared with its general Tolkien—non-weighted occurence frequency.

Relevance

Filter books with positive ranking gain $\kappa - \kappa' > 0$, and plot a scatter plot

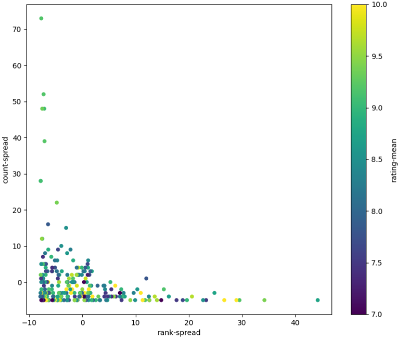

Figure: Scatter plot of Tolkien co-rated book metrics

Note that we could select books for further analysis in different ways. Prefering those with highest rank gain, those with high count, or some sort of combined multi-objective (Pareto) selection.

Linking

Build a graph of links between books with positive ranking gain:

- create set of most common Tolkien books $\{\tau_k\}$ and rank gain books $\{\rho_l\}$, $\{\mu_m\} = \{\tau_k\} \cup \{\rho_l\}$
- count pair-wise co-ocurrence in the same rating group γ_i for each pair (μ_m, μ_n)

Getting linked group

Estimate a group of books linked to LOTR books and to one another:

- create a normalized vector of weights $\tau^{(1)}$ as relative LOTR books count
- compute a vector of non-LOTR books linked (ρ—τ interactions) to this vector, $\rho^{(1)}$
- redistribute weights $\rho^{(n-1)} \to \rho^{(n)}$ by links between the books while weight std grows (ρ—ρ interactions)
- select the top books from $\rho^{(n)}$

$$\begin{bmatrix} \rho^{(1)} \\ 0 \end{bmatrix} = \left[\begin{array}{c:c} \rho\text{—}\rho \text{ interactions} & \rho\text{—}\tau \text{ interactions} \\ \hdashline \tau\text{—}\rho \text{ interactions} & \tau\text{—}\tau \text{ interactions} \end{array}\right] \begin{bmatrix} 0 \\ \tau^{(1)} \end{bmatrix}$$

$$\left[\rho^{(n)}\right] = \left[\rho\text{—}\rho \text{ interactions}\right]\left[\rho^{(n-1)}\right]$$