Learning to Place Objects using Networks

Xindi Wang[1,2], Onur Varol[1,2], Tina Eliassi-Rad[2,3]

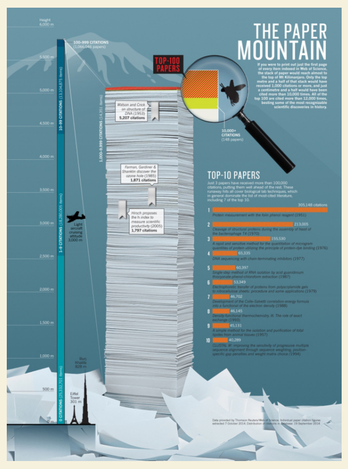
1. Center for Complex Network Research, Northeastern University, Boston, USA
2. Network Science Institute, Northeastern University, Boston, USA
3. College of Computer and Information Science, Northeastern University, Boston, USA

Predict Heavy-tailed Attributes of Items

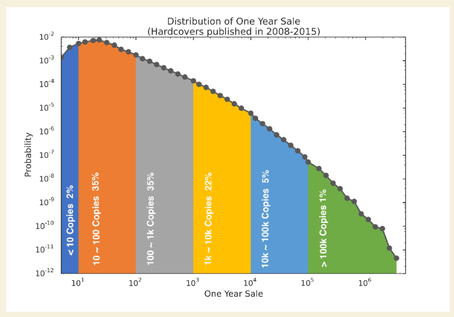
There are many items with heavy-tailed attributes in the real world. Examples include paper citations, book sales, and meme cascade-sizes. Most of the time, we are interested in the items at the high-end of these heavy-tailed distributions: papers with many citations, bestselling books, and "viral" memes.

A heavy-tailed distribution naturally leads to a data imbalance problem because we have far more items at the low-end of the distribution. Thus, traditional methods like Linear Regression underpredicts the attributes of items at the high-end of the distribution.

Paper Citations



Book Sales



Meme Cascade Size

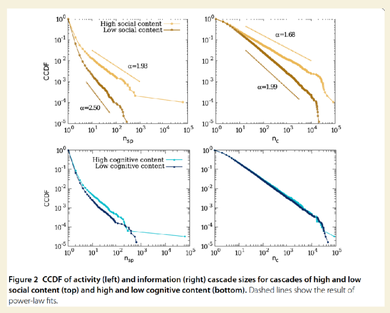
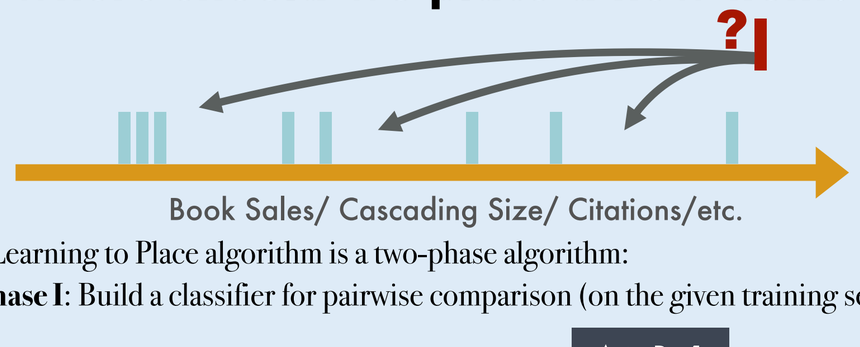


Figure 2 CCDF of activity (left) and information (right) cascade sizes for cascades of high and low social content (top) and high and low cognitive content (bottom). Dashed lines show the result of power-law fits.

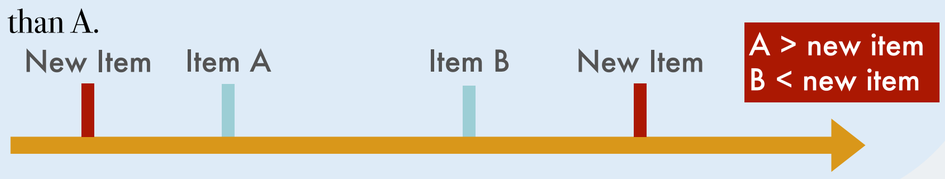
Learning to Place: Given a Sequence of Ordered Items, where should we place a new item?



The Learning to Place algorithm is a two-phase algorithm:

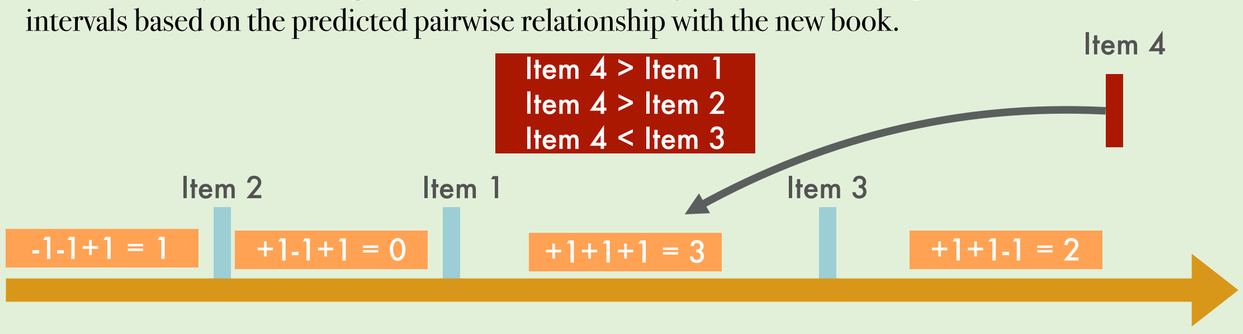
- **Phase I**: Build a classifier for pairwise comparison (on the given training set)

- **Phase II**: Find places for new items based on the predicted pairwise relationships. **This is non-trivial because the classifier from Phase I may output conflicting results.**
  - Example: Phase I classifier predicts that A is better than a new item and B is worse than a new item. This conflicts with the fact that B is better than A.



Learning to Place: Resolving Conflicts

Method 1: Majority Voting. Each book in the training set is a voter and upvotes/downvotes on intervals based on the predicted pairwise relationship with the new book.



Majority Voting algorithm example. Item 4 is better than Item 1, therefore intervals on the right of item 1 would gain one positive vote and on the left of item 1 will gain one negative vote. Same process for item 2 and item 3. After all trained books voted, we select the third interval because it obtained the most votes; and therefore we would place item 4 in the third interval.

Method 2: Weighted Tournament Graph (WTG) wave. A weighted tournament graph has each node as an item and if there is an edge from item A to item B with weight w, it means that item A "beats" item B with confidence w. Therefore items with high out-degree (weighted) and low in-degree (weighted) would have higher rank.

A beats B with confidence w High Out-degree, Low In-degree High In-degree, Low Out-degree

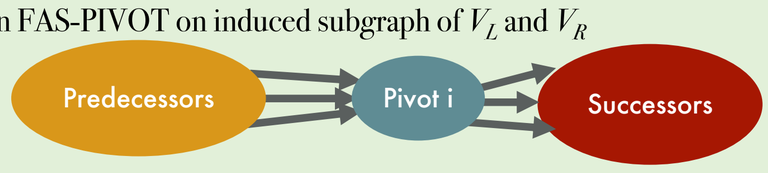
We propose a method which extends on Cohen et al.[1] where they would iteratively remove the node with highest potential (defined as outdegree minus in-degree) to obtain a ranking. However, Cohen et al.'s method is rather local. To enhance it, we not only look at the potential of the node itself but also the potential for its successors. We define the score as:

$$\text{score}(i) = \alpha \cdot \text{potential}(i) + (1-\alpha) \cdot \sum_{j \in N_i} \text{potential}(i)$$

At each step we select and remove the node with the highest score and then recalculate the score for all nodes in the remaining network.

Method 3: FAS-PIVOT [2]. FAS-PIVOT is an approximation algorithm to solve minimum feedback arc set problem. The algorithm is iterative, where at each step we:

- Randomly pick a pivot node i
- Put i's predecessors in V_L and i's successors in V_R
- Recursively run FAS-PIVOT on induced subgraph of V_L and V_R

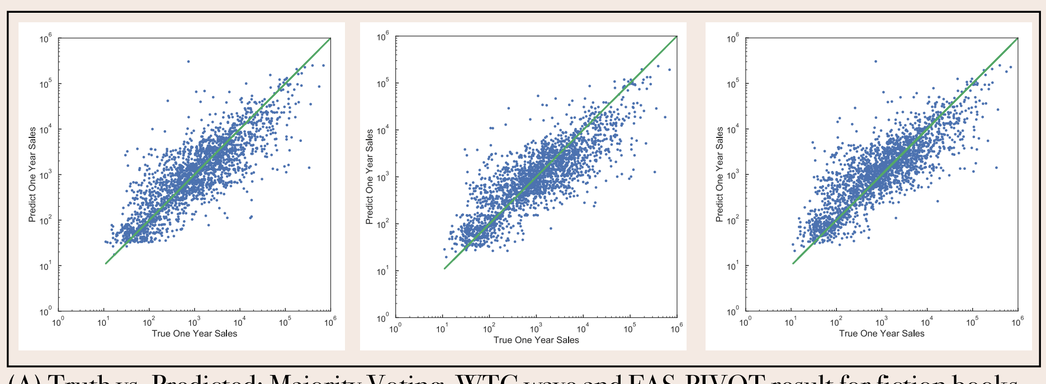


Method 4: SpringRank [3]. SpringRank is a physically-inspired model and algorithm to infer hierarchical rankings of nodes in directed networks. Each edge is regarded as a spring and the objective is to minimize the energy of the system.

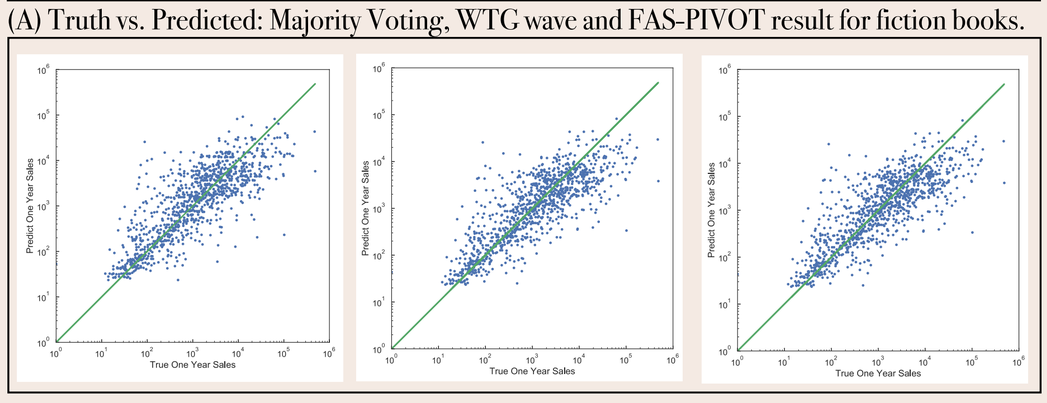
Data

We have a book data set. For each book we have 55 features such as the fame and previous sales of the author, the publisher value of this book, the topic of the book, etc. We tested our algorithm on 2,683 fiction books and 1,111 biography books.

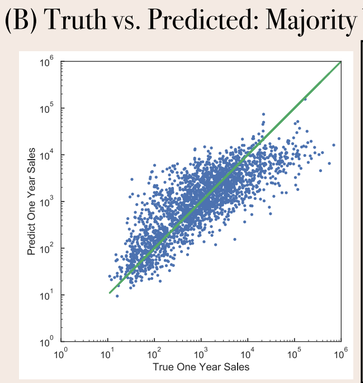
Results



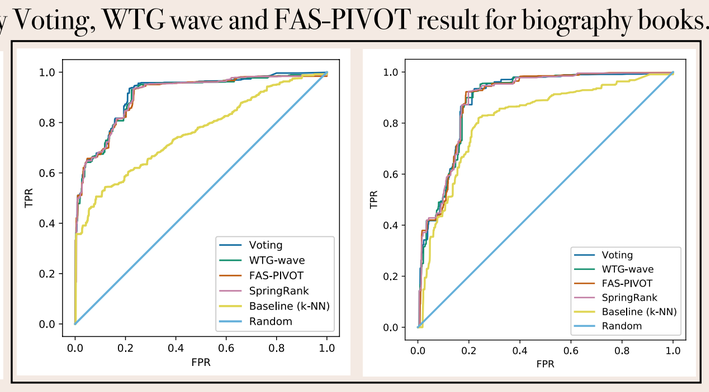
(A) Truth vs. Predicted: Majority Voting, WTG wave and FAS-PIVOT result for fiction books.



(B) Truth vs. Predicted: Majority Voting, WTG wave and FAS-PIVOT result for biography books.



(C) Truth vs. Predicted: Linear Regression on fiction books. Linear regression has systematic underprediction at the high end when the true one-year sales is more than 10,000.



(D) Fiction and biography ROC curve of various algorithms for learning-to-place and the K-nearest neighbor baseline. We see that different learning-to-place methods have very similar performances; and they all outperforms the K-nearest neighbor baseline.

References

[1] Cohen, William W., Robert E. Schapire, and Yoram Singer. "Learning to order things." Advances in Neural Information Processing Systems. 1998.
[2] Ailon, Nir, Moses Charikar, and Alantha Newman. "Aggregating inconsistent information: ranking and clustering." Journal of the ACM (JACM) 55.5 (2008): 23.
[3] De Bacco, Caterina, Daniel B. Larremore, and Cristopher Moore. "A physical model for efficient ranking in networks." arXiv preprint arXiv:1709.09002 (2017).

Xindi Wang: xindi.w1993@gmail.com

Northeastern University Northeastern University Network Science Institute