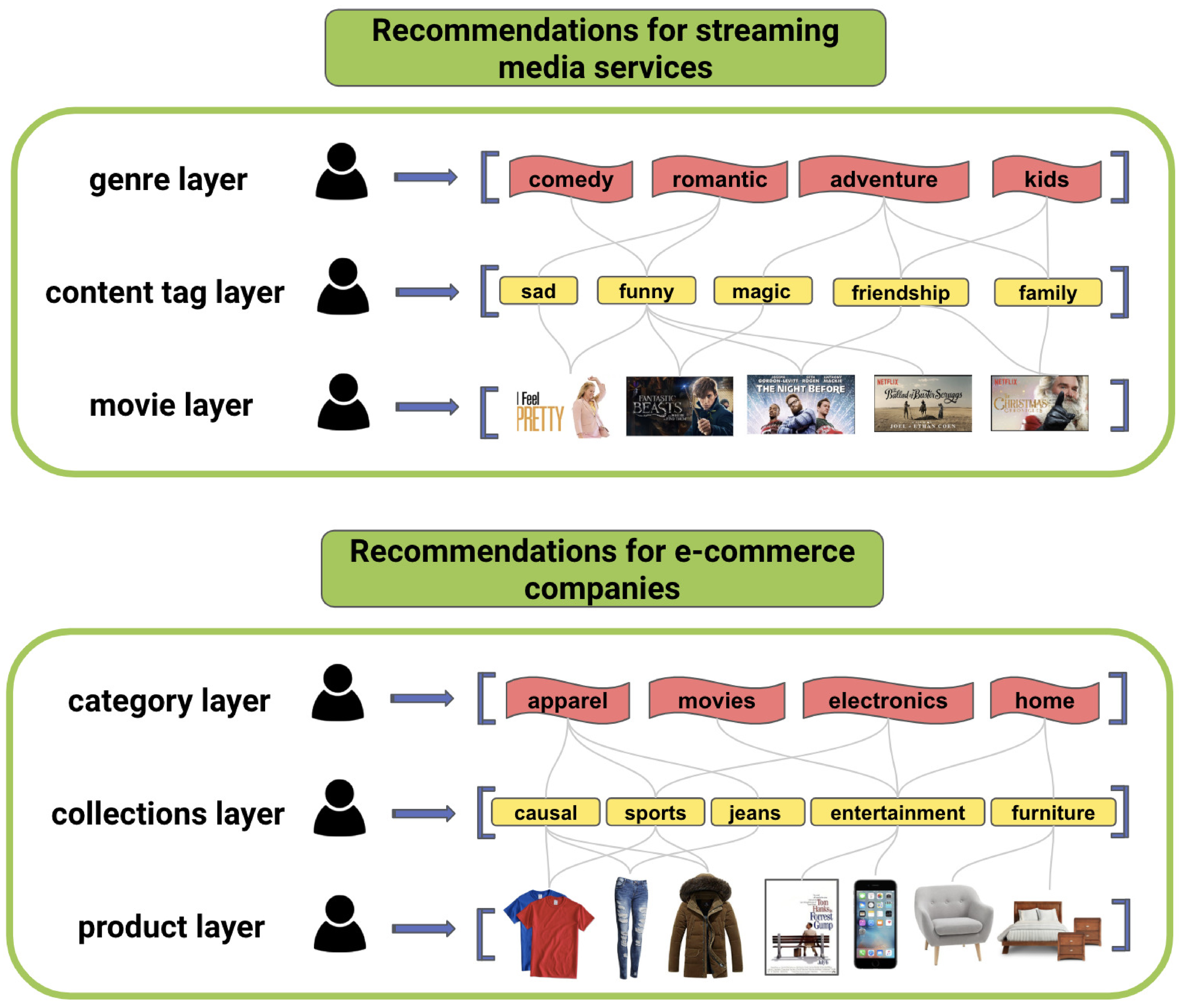
Collaborative Autoencoders for Top-K Hierarchical Recommendations

Dora Jambor • Putra Manggala

Shopify Inc., Montreal, Canada

Motivation

In real world recommender systems, items can be organized into hierarchical taxonomies, e.g. a movie can be characterized by a set of content tags, and a content tag can belong to multiple genres. Top-K hierarchical recommender systems generate top-K recommendations for each layer in a hierarchical item taxonomy.



In this work, our objective is to answer the following key questions:

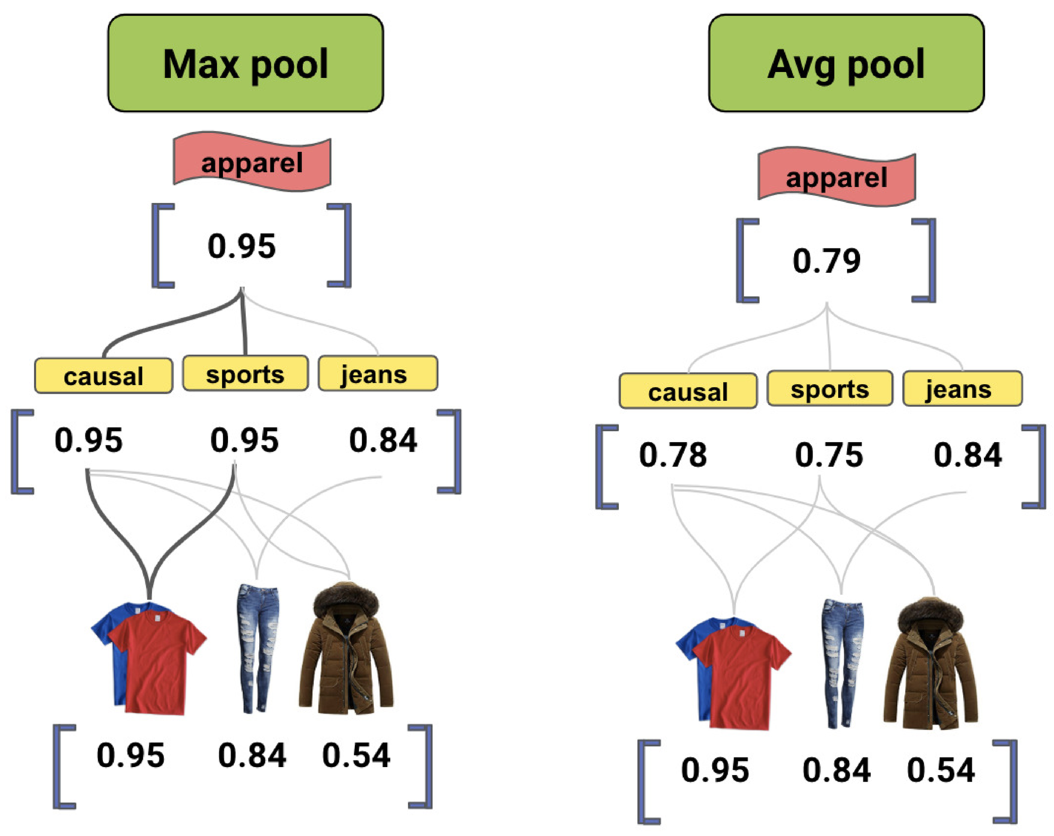
◇ Can we solve all the top-K recommendation tasks by training **one** model?

◇ Can we improve model performance by directly incorporating the item taxonomy labels into our objective function?

Baseline Approaches

◇ **Approach 1:** Build one recommender system per taxonomy layer.

◇ **Approach 2:** Estimate top-K item recommendations, then retrieve top-K tag and category recommendations by performing max or average pooling on the item-layer predictions, as follows:



After pooling, we sort prediction scores at each taxonomy layer and serve the highest K scoring items as top-K recommendations.

Top-K Hierarchical Recommendations

Our method incorporates losses due to misclassifications from all layers into one loss function. Using an autoencoder model with a hierarchical loss function (inspired by [1][2]), we first compute bottom layer output predictions:

$$p_{m=1} = \mathrm{autoencoder}(v), \tag{1}$$

where v is the appropriate vector representing a user's past item interactions, and m denotes the taxonomy layer, e.g. m=1 could refer to the movie layer.
Predictions for $m > 1$ are then derived by:

$$p^k_{m+1} = \underset{j}{pool}\, p^j_m, \tag{2}$$

where the taxonomy node j at layer m is the child of taxonomy node k at layer m+1 and for pooling, we use max or avg pooling. We then use the cross-entropy loss at each taxonomy layer:

$$J_m = -\sum_{k=1}^{C_m} \left(y^k_m \log p^k_m + (1 - y^k_m) \log (1 - p^k_m) \right), \tag{3}$$

where C_m is the number of nodes at layer m. The total loss is then written as:

$$J = \sum_{m=1}^{M} \omega_m J_m, \tag{4}$$

where M is the maximum taxonomy level, and ω_m is a hyperparameter to control layer m's loss.

Experimental Setup

The following datasets will be used for evaluation:
(i) **Shopify dataset** of users' app installations, where each app is mappable to tags and tags to categories;
(ii) **MovieLens 20m dataset** that contains users' ratings of movies where movies are classified into genres and user-assigned tags.

Datasets	users	items (C_1)	density	C_2	C_3
Shopify apps	9,085	2,342	0.48%	142	12
ML20m	114,050	15,578	0.17%	36,947	20

The performance will be measured by an aggregate MRR metric. For each taxonomy layer m, we compute Mean Reciprocal Rank for layer m (MRR_m):

$$MRR_m = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{1}{rank_{m,u}} \tag{5}$$

where U is the number of users and $rank_{m,u}$ is the position of the first relevant item for user u in taxonomy layer m. The single aggregate metric is then computed by a convex combination of MRR_m from all layers. The convex combination parameters can be guided by product or marketing intentions.

Bibliography

[1] Xiong, T., Manggala, P. (2018). Hierarchical Classification with Hierarchical Attention Networks.

[2] Sedhain, S., Menon, A. K., Sanner, S., Xie, L. (2015, May). Autorec: Autoencoders meet collaborative filtering. In Proceedings of the 24th International Conference on World Wide Web (pp. 111-112). ACM.