# Online Learning for Distributed and Personal Recommendations

## A Fair Approach

**Martin Tegnér**

1  Recommendations

A recommendation system presents items (80257360) to a user. The aim is to make relevant suggestions, for a product purchase, friend request or movie watch. **Supervised learning** trains a map that recommends an item y from user inputs x. This can be seen as a graph, where the model M connects all items based on x. The **shortest edge** from the current item yields a recommendation.

**Personalised** models are expressive but **data hungry** to be able to generalise: informative features encoded in x typically rely on personal and private information, and model training on centralised storage of user data. On the other hand, **contextual** graphs can be constructed from item features, or inferred from sequences of item-views alone.

This work presents a method **not relying on private user information and data collection**, still providing **personalised** recommendations.

2  Probabilistic Mixture Model

At the base, we assume a collection of graphs, representing different (contextual/inferred) activities, i.e. relevant ways of relating items for recommendations. We model each with a Markov chain. A high (low) probability \( p(y|x) = A_y \) yields a strong (weak) relation between item y and x, where A is the chain’s transition matrix.

For a model both expressive of different activities and adaptive to individual behaviours, we use a latent categorical variable \( C \in \text{Cat}(n) \) to mix the Markov chains at user level: \( p(C|k) = \pi_k \) is the probability of kth activity with items distributed by Markov chain \( A^{(k)} \).

Let \( y_t = (\ldots, y_{t-2}, y_t, y_{t+2}, \ldots) \) be user-viewed items at iteration t. If \( L(y_t) \) is likelihood under chain k, we have that the posterior \( p(y_t|C) \) remains a \( \text{Cat}(n) \) with \( \pi(k) = Z^{-1}L(y_t) \pi_k \). For new data \( y_{t+1} \), Algorithm 1 updates the posterior in an online and distributed way:

1. only \( \pi_{\text{new}} \) is needed for computing \( \pi \) of the new posterior, and (2) inference is distributed using user data locally, at the user’s device. Further, the posterior predictive \( p(y_t|\pi_{\text{new}}) \) for recommendations is distributed.

3  Experiments

We construct three activities for 300 home-furnishing products:
- \( A^{(1)} \) a functional context, according to (i) kitchen (100 items), (ii) living-room (100), and (iii) bathroom (100). See figure below.
- \( A^{(2)} \) an inspirational context, extracted from product images designed by home-furnishing specialists.
- \( A^{(3)} \) an add-to-cart activity, with \( A^{(3)} \) estimated from item-sequences of order data.

**The functional shopper.** We simulate 100 items from \( A^{(1)} \) and feed the algorithm with two items for each posterior update.

Resulting \( \pi(t) \) (left). Weights stabilise after a few iterations, with a high probability for the generating functional activity. t-SNE visualisation of transitions generated by \( \pi(t) \) (below) along with posterior recommendations.