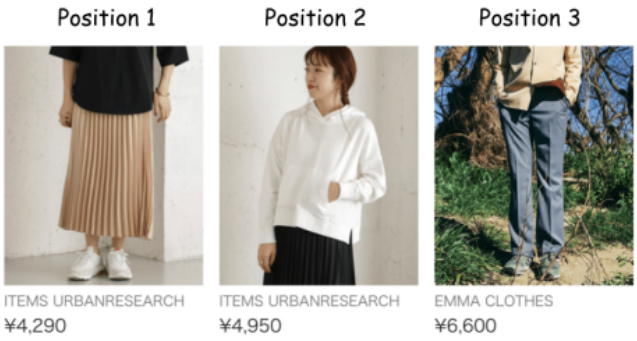
Group Assignment
Deep Q-learning for Recommender Systems

Learning objectives

- Learn how to apply Deep Q-learning to learn a recommender system.
- We do not have a simulator or access to the business environment to return s' and r, so balancing exploration and exploitation and action selection is not part of the assignment; the focus is on the design of the MDP and training the DQN.
- Understand how the data generating process impacts learning Q-values.

Data:

ZOZO Inc. is the largest e-commerce company in Japan. It uses MABs to recommend fashion items to users on their platform, called ZOZOTOWN. The data was collected in a 7-day experiment in late November 2019. During the campaign, for each user impression, one of two policies was randomly selected to select a recommendation: a random policy, and a Thompson Sampling policy. Each policy selects **three** of the candidate fashion items for each user, and in which position to place these three items in the recommendation interface. The figure below shows that the customer would see.



There are two sets of data, one generated by a random policy and one generated by Thompson sampling. Each set has two files: *all.csv*, and *item_context.csv.*

all.csv
Each row of the data has features vectors such as age, gender and past click history of the users. These feature vectors are hashed to avoid sharing personally identifiable information.

- timestamp: timestamp of impression
- item_id: index of items as arms (index ranges from 0-80).
- position: the position of an item being recommended (1, 2, or 3 correspond to left, center, and right position of the ZOZOTOWN recommendation interface, respectively).
- click: target variable that indicates if an item was clicked (1) or not (0).
- propensity_score: to ignore.

- user feature 0-4: user-related feature values, hashed.
- user-item affinity 0-: user-item affinity scores induced by the number of past clicks observed between each user-item pair.

item_context.csv
Each row of data includes item-related features such as price, fashion brand and item categories.

- item_id: index of items as arms (index ranges from 0-80).
- item feature 0-3: item related feature values

Deliverables:

Code (.py file)
-Code to prepare the data and learn a DQN
-The code needs to be run on two different datasets: data generated by a random policy and data generated by Thompson sampling.
-The main goal is to compare how fast the Q-values converge if we use data generated by a random policy versus data generated by Thompson sampling.

Presentation
-Details on code and training
-Useful insights
-Maximum 7 min