Introduction
In this paper, we propose a recommender system for the Scratch online community. Scratch users create, share and remix projects by using the Scratch programming language developed by the Lifelong Kindergarten Group at the MIT Media Lab. The proposed recommender system utilizes project tag information to determine similarities between various users and then uses these relationships to identify the optimal set of items to be recommended. Our aim is twofold:
(i) Create and evaluate recommendations based on two different types of input tags (explicit and implicit) and
(ii) Evaluate recommendations based on relevancy and diversity.

We are including diversity in our recommendation algorithm under the assumption that recommending a more varied set of items will be more valuable to users than simply recommending similar items. Through a calculated combination of relevancy and diversity, our recommender system is aimed at leading users to explore further into the Scratch community and improving the productivity of “passive producers” by using the output of “active consumers”.

Methods

Two different inputs:
1) User content tags; tags of the projects created by the user (shared tags)
2) User favorite tags; tags of the other users’ projects that the user has explicitly favorited.

The algorithm:
1) Using latent semantic indexing (LSI), we constructed a collaborative view of user profiles and created user-tag and user-user similarity matrices in a reduced dimensional space.
2) Using MMR algorithm, \(\text{MMR}(Q, Di) = \lambda \text{Sim}(Q, Di) + (1 - \lambda) \text{max Sim}(Dj, Di)\), we iteratively selected the users with highest similarity to our query and then updated the remaining user similarity scores by computing the degree of dissimilarity between each user and the previously selected ones.

The challenge for our recommender system is to balance this tradeoff between two possibly conflicting objectives of:
1) recommending highest ranked similar users and
2) recommending highly diverse (novel) ones.

In order to find the best balance, we conducted three sets of controlled studies on Amazon Mechanical Turk.

Controlled Studies

Conditions:
C1. We evaluate the shared tags of randomly selected users and \(\lambda\) values of 0, 0.5 and 1. To test the quality of our final recommendations, a set of HITs were posted on Amazon Mechanical Turk. Each HIT was opened to 20 Amazon Mechanical Turk workers. Workers were presented with the project tags of our users and recommended users’ project tags. They were asked to review the recommendations and indicate their likelihood of viewing recommended projects.
C2. We evaluate the favorite tags of our users and \(\lambda\) values of 0, 0.5 and 1. A new set of HITs was posted for condition 2 in the same format of condition 1.
C3. We randomly selected 20 users from our user pool. 10 were assigned to the shared tags group and 10 were assigned to the favorite tags group. Turkers were asked to evaluate these users similar to conditions 1 & 2.

Results:

We compared the performance of the recommender system in regard to the input of the system. When viewing the recommended user’s projects, the favorite tags group performs significantly better than the shared tags group (Table 3). When compared with the control group, the recommendations based on shared tags do not perform well (Table 4). However, compared with the control group the recommender system performs better when using favorite tags (Table 5).

Conclusions
Traditionally, recommender systems focus on generating recommendations that are very similar to users’ interests. This can result in too analogous recommendations that fail to engage the users. Hence, this paper proposes a tag-based recommender system that generates diversified recommendations for Scratch online community.

Our initial experimental results show that recommendations that balance the notions of relevancy and diversity \((\lambda = 0.5)\) perform better.

Our results also show that using favorite tags instead of shared tags may result in better recommendations. The act of “favoring” might be a better indication of user interest and show a clearer direction to what future projects user would like to create next. In that sense, a favorites list could be used as a predictor and a repository for future project ideas.

Future Research
Several things can be done to improve the recommender system for future studies. Different synthesized or natural datasets could be used to test the recommender algorithm. Using a dataset that Mechanical Turk workers are more familiar with (movies, books etc.) would help us better translate the recommendations onto the AMT platform. Additionally, a genetic algorithm can also be used in the optimization of the final recommendation set.

Another study we are interested in conducting is to generate Scratch project recommendations based on the similarities of the project codes. As assigning tags is not a requirement on Scratch, users are inconsistent when tagging their projects and as a result, our current recommender system would not be useful in those cases. Since all projects have codes and we are able to get information on remixes, project codes can be used for user profiling. A third study we might take on is developing a social matching system for Scratch. Since Scratch is an online collaboration community, it would make a good platform for a social matching system that aims to increase social interaction and foster collaboration.

Acknowledgements
This material is based upon work supported by the National Science Foundation under Grants IIS- 0968561 and IIS-1211084. Scratch is a virtual community created and managed by the MIT Media Lab.