Prastava: An open-source Ruby-based generic recommendation system

Min-Yen Kan[1]**, Himanshu Gahlot[2], and Tarun Kumar[3]

[1] Department of Computer Science, National University of Singapore, Singapore
[2] Motilal Nehru National Institute of Technology, Allahabad, India
[3] India Institute of Information Technology – Allahabad, India
kanmy@comp.nus.edu.sg and {himanshu.gahlot86,tar.iiita}@gmail.com

Making recommendations for patrons based on circulation records and social tagging has become a service that libraries have increasingly needed to implement to expand their services. To address this need, we have released Prastava[4], an open-source recommendation system that aims to fill this gap. Prastava ("recommendation" in Hindi) is an LGPLed recommendation system that is written purely in the Ruby programming language, which makes it suitable for both commercial and non-commercial web application frameworks. Prastava incorporates Ferret, an open-source Ruby text search engine and runs in a client-server mode for most efficient recommendation generation.

Prastava integrates the two popular forms of recommendation: collaborative filtering (CF) and content-based filtering (CBF). User-based CF relies on past recommendations of users on items to calculate which users rate items in a similar manner to a target user. In contrast, CBF uses descriptors gleaned from the items themselves to determine item similarity. As examples, for recommending webpages, the words on the page could serve as the content; for recommending movies, the metadata (genre, cast, director) could serve. Prastava integrates the English language Porter stemmer and TF×IDF weighting scheme, both popularly used in search engines, to serve in its CBF subsystem. Both recommendation methods rely on the calculation of a similarity metric; Prastava lets the developer choose between cosine similarity and Pearson's correlation.

CF works well in cases where items have been rated by many users, but fails on items without prior ratings. In contrast, CBF overcomes this "cold-start" problem but has been shown to underperform CF. Thus, Prastava also implements a hybrid recommendation scheme that combines both CF and CBF recommendations, based on the number of users who have rated the candidate item. The developer can choose between CF, CBF and hybrid easily by changing the configuration files for the client and server.

Prastava's project page incorporates two collaborative filtering demonstrations. In the first, Prastava uses a sizable slice of data from the MovieLens project (1663 titles, 942 users and 100,000 ratings). In the second, we allow the user to construct their own toy dataset to try out the CF system.

** This work was supported by a Media Development Authority (MDA) grant "Interactive Media Search", R-252-000-325-279.

[4] Available off of http://wing.comp.nus.edu.sg/downloads/