In this demo, we present a hybrid system which combines long term user profile, the current context describing the last visit and the most popular story. The long term user profile is built based on the user’s long term browsing habits. The context is described by the user’s local time and location, last read story and referring link to the last read story. Based on these features, the system predicts the most likely news categories of interest to the user. From the top three predicted categories it recommends the most popular news stories appearing in the last six hours. We show that the system can outperform several base-line systems: most popular, item-to-item collaborative filter, and contextual. Furthermore the system is running live providing real-time recommendations of news articles to a large user base.

2 Methodology

The proposed methodology consists of several steps. (1) First, we analyzed the user base data and split it into two groups: “old” users which have a history of more than 50 visits and, “new” users which have a history of less than 50 visits (the threshold 50 was selected based on preliminary experiments). History giving a long term user interest is included in feature representation for “old” users only. (2) Then, a separate model is trained for each of the two user groups. To avoid overspecializing, we define a machine learning problem as predicting the most interesting news category rather than specific news articles. In our specific setting the news stories were manually classified by domain experts into a taxonomy of 40 categories. Alternatively, one can use machine learning to automatically classify the articles into taxonomy.

Experimental evaluation was conducted using different combinations of feature sets (as input for the SVM) to predict the top three categories of news articles the user would be interested in. Table 1 shows that, for “old” users, the precision\(^2\) using context information, such as referring page and time is relatively low (36%). Including requested page is more predictive (41%), while using history information gives the highest precision (48%). However, if we include all the features (i.e. history, Geo, requested and referring page and time), we obtain the highest precision (52%).

<table>
<thead>
<tr>
<th>All</th>
<th>History</th>
<th>Geo</th>
<th>Requested</th>
<th>Referring</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>52</td>
<td>48</td>
<td>43</td>
<td>41</td>
<td>36</td>
<td>36</td>
</tr>
</tbody>
</table>

Table 2 shows that our system can do good predictions even for “new” users, and that the quality of the predictions using all the features again gives the highest precision.

(3) After predicting the top 3 categories, the system selects the top new article in each category and suggests this set of news to the user. This approach provides robustness and diversity since we are only predicting interesting categories for each user and are not linking him/her directly to stories.

\(^2\) From the categories the SVM model predicts for user A, which were actually interesting enough so user A read an article from that category in the next step.
Table 2. Precision for sets of features for “new” users (less than 50 visits) on top 3 categories

<table>
<thead>
<tr>
<th>All</th>
<th>Geo</th>
<th>Referring</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>45</td>
<td>36</td>
<td>35</td>
</tr>
</tbody>
</table>

We deployed our system (see Fig. 1) in real-world setup with millions of requests and hundreds of new articles per day. In our server logs each user visit and updates the index in real-time (batches of 1000 visits). The logs are also archived and, along with the rest of the data, used daily for training new SVM models. The co-visitation matrix of the collaborative filtering module is updated on the fly. Newly published stories are crawled as soon as they get the first visit. The average time to compute and serve the recommendation is 20 ms³.

![Real-time recommender system architecture (only SVM and CF modules represented)](image)

We compared the performance of our system against 4 other approaches (1) item-to-item collaborative filtering based co-visitation matrix (users who read this, also read that), (2) random category which selects three random categories and recommends the most popular stories in these categories (3) most popular which recommends the top most popular stories at the moment of the visit, and (4) contextual which recommends based on the last read story. Fig. 2 presents the results over 2 weeks, it shows that our system outperformed all the others in terms of transition probabilities. This means that, provided with the set of recommended news articles, the user is most likely to click on the articles recommended by the SVM module. Random category and collaborative filter modules performed on par over the two weeks of the trials. The third ranked was most popular and the fourth was contextual.

3 Overview of the Demonstration

The demonstration will consist of a system running on a large live feed of news articles and page-views with user profiles and visit information being updated on the

---

³ This may vary according to location and connection speed. The reported value is an average of loading times reported by Chrome.