

Serendipitous Recommendation for Mobile Apps Using Item-Item Similarity Graph

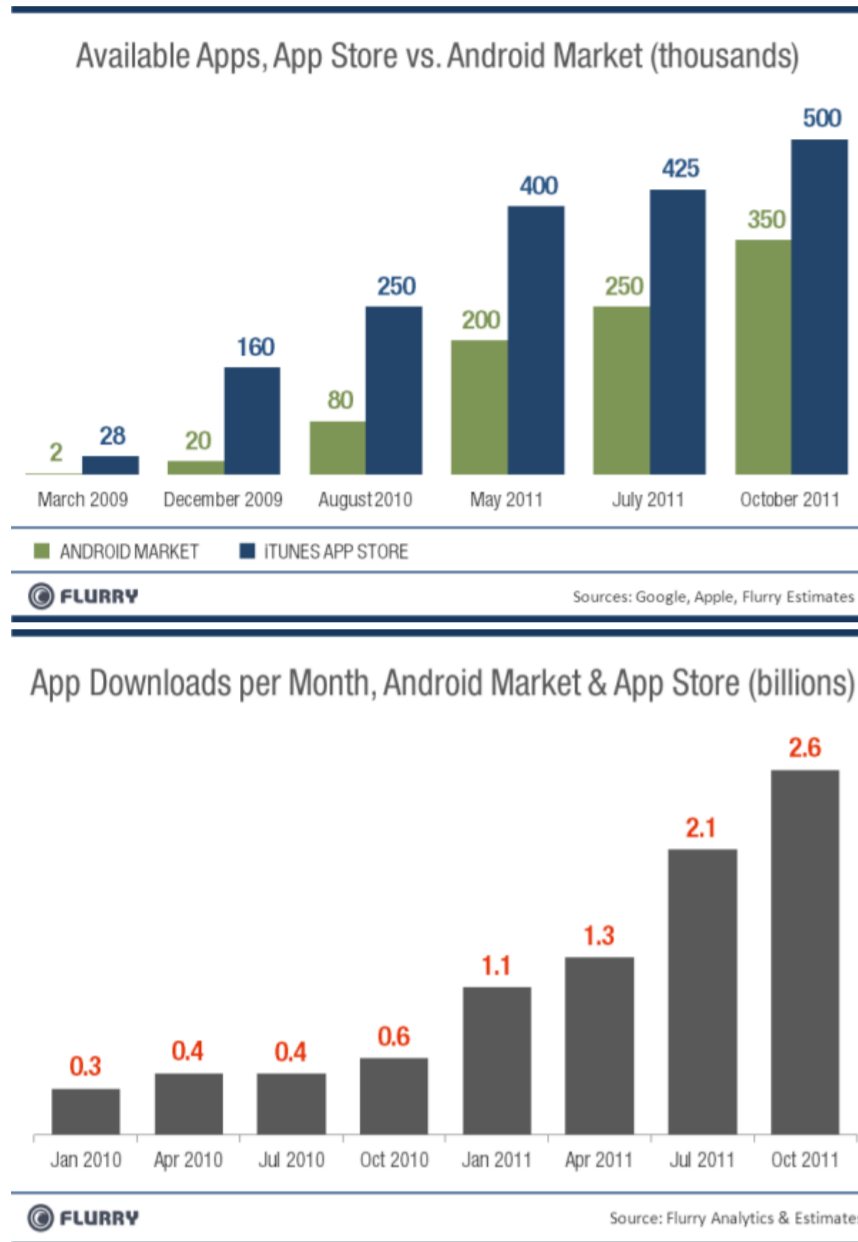
Upasna Bhandari¹, Kazunari Sugiyama¹, Anindya Datta¹ and Rajni Jindal²

¹School of Computing, National University of Singapore

²Department of Computer Science, Delhi Technological University

{a0106246, sugiyama, datta}@comp.nus.edu.sg, rajnijindal@dce.ac.in

❖ Background



Information Overload for User



❖ Problems of Existing Recommendation Systems

- The existing recommendation systems have a key assumption, "every item must be used at least once and every user must use at least one item"
- To illustrate, in Amazon.com, after browsing a couple of items, we are provided with lists such as *Recommended For You* or *Customer Who Bought This Item Also Bought*
- Looking at them closely, users often observe that all of these items are already known. This may not be ideal for overall user satisfaction and experience with the system
- For example, if a user browses a book written by *Dan Brown*, most of the recommendations for the user will be books by *Dan Brown*

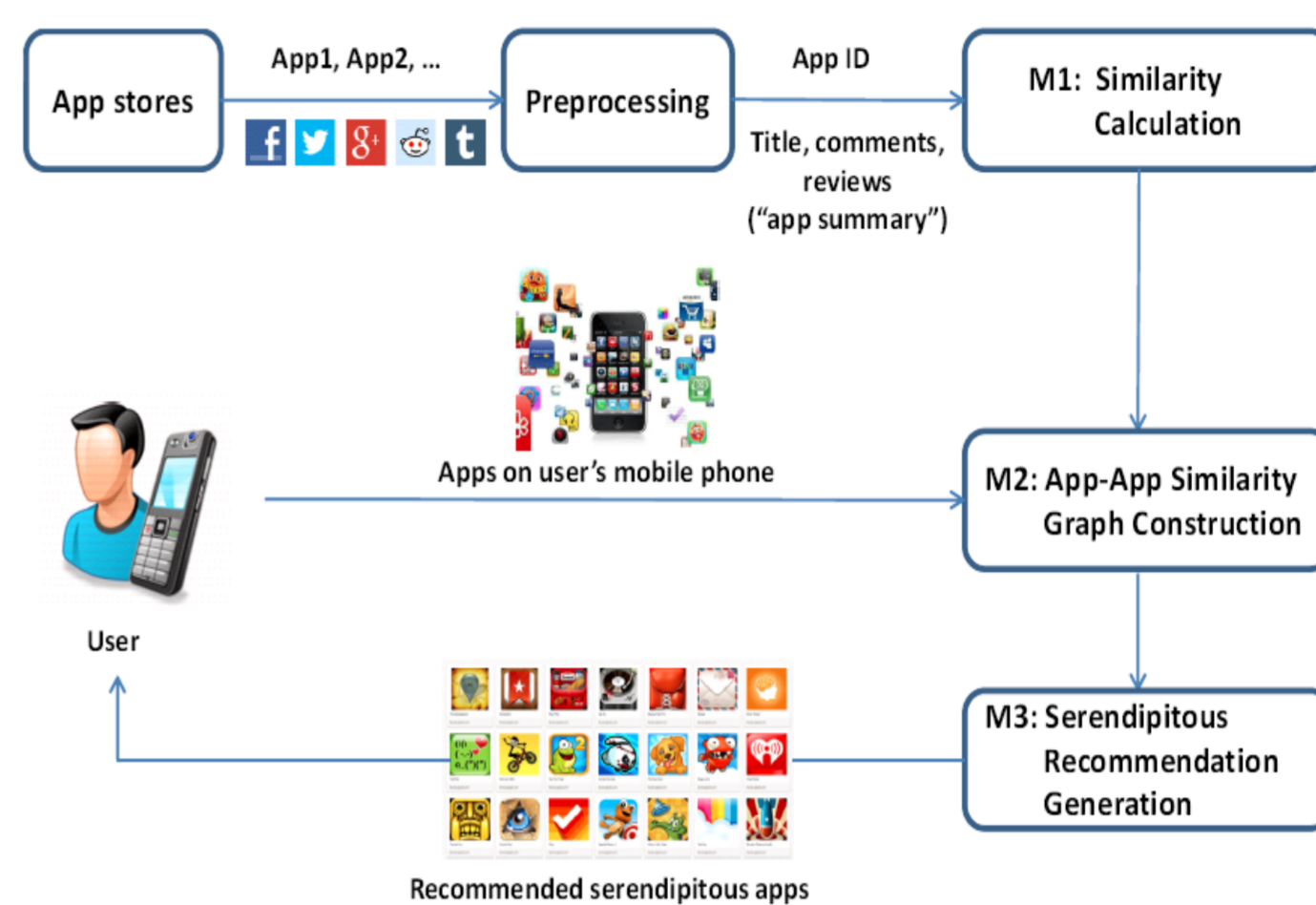
It is reasonable to say that a user would be happy with recommendation systems that offer less obvious choices.

❖ Related Work

- Serendipitous Recommendation**
Ziegler et al. (WWW'05), Andre et al. (CHI'09), Kawamae (SIGIR'10), Nakatsuji et al. (CIKM'10), Sugiyama and Kan (JCDL'11)
- Recommendation for Mobile Apps**
Yan and Chen (MobiSys'11), Yin et al. (WSDM'13), Lin et al. (SIGIR'13)

"Our research is the first work on serendipitous recommendation for mobile apps"

❖ Proposed Method



Given a set of apps installed on a users' phone, our approach generates serendipitous recommendation for mobile apps based on the users' installed apps

M1: Similarity Calculation between apps represented by TF-IDF

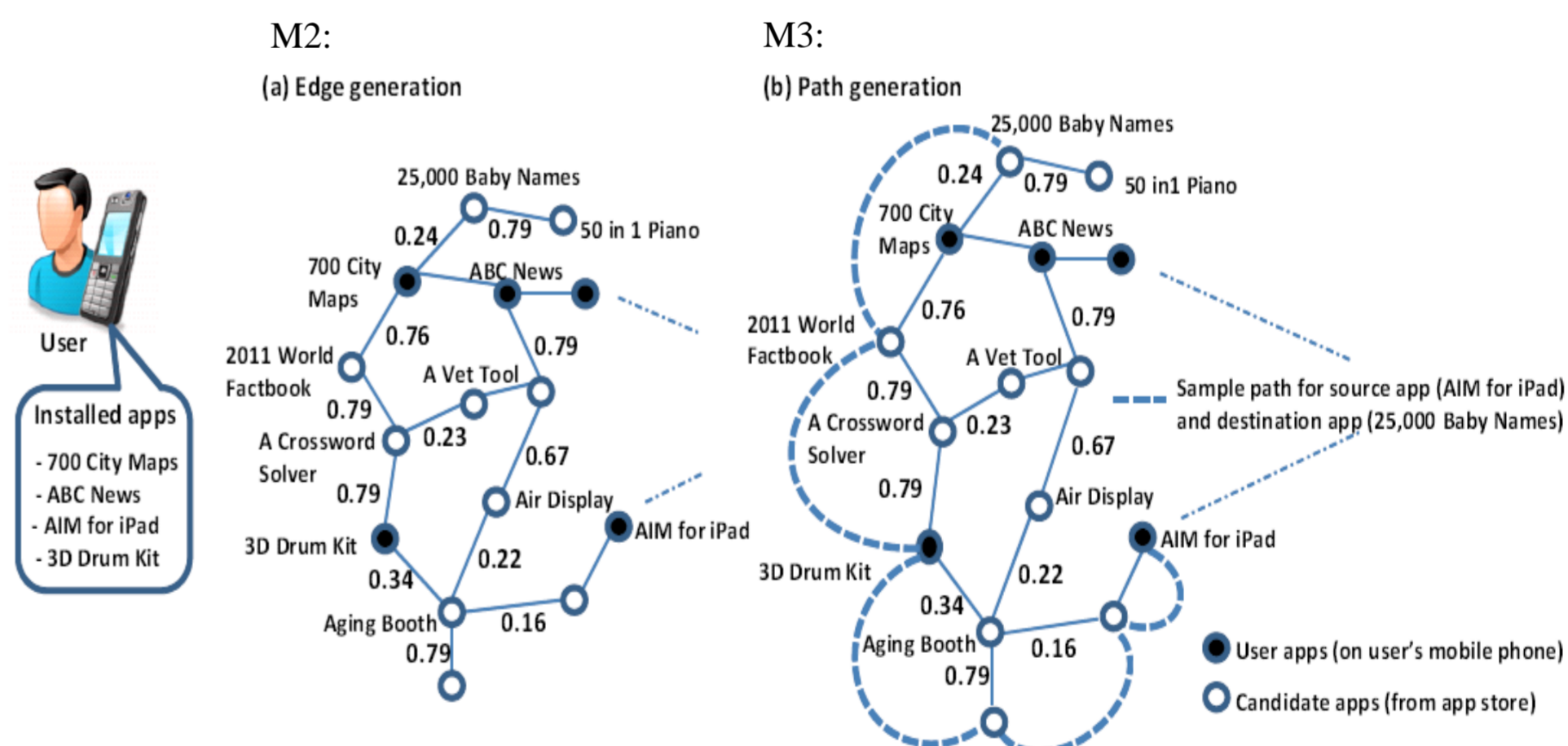
- Consider all app pairs and compute similarity scores between them
- $$sim(f^{a_i}, f^{a_j}) = \frac{f^{a_i} \cdot f^{a_j}}{|f^{a_i}| |f^{a_j}|}$$

M2: App-app Similarity Graph Construction

- Vertices: Apps
- Edges: Similarity scores between the vertices

M3: Recommendation Generation

- Use list of apps in a users' phone to construct paths from one app to another considering each app pair as source and destination
- Shortest-path algorithm



❖ Experiments – Recommendation Accuracy

- Experimental Data**
66,223 apps and 22,213 users, collected from Apple iTunes, Google Android, Windows and Blackberry
- Evaluation Measure**
Normalized item novelty (nITN), diversity-in-top-N

