



# Personalized Mobile Application Discovery

Bo Yan and Guanling Chen

Department of Computer Science  
University of Massachusetts Lowell

# How to Find Apps

- Search by keywords or browse by categories
- **Personalized recommendation** according to user download history or ratings
  - Users often install apps to try them out without uninstalling when dislike them
  - Lack of ratings
    - Angry Birds, ratings from 7% downloads
  - Outdated ratings for continuously updated apps

# AppJoy

- Usage Score
  - Implicitly measure how the apps are used by users without requiring explicit ratings
  - Adaptive to the changes of user taste
- Collaborative Filtering
  - Compute similarity by usage score
  - Predict user preference from the similar apps of user's installation

# Talk Outline

- Recommendation
- Implementation
- Evaluation Results
- Conclusion and Future Works

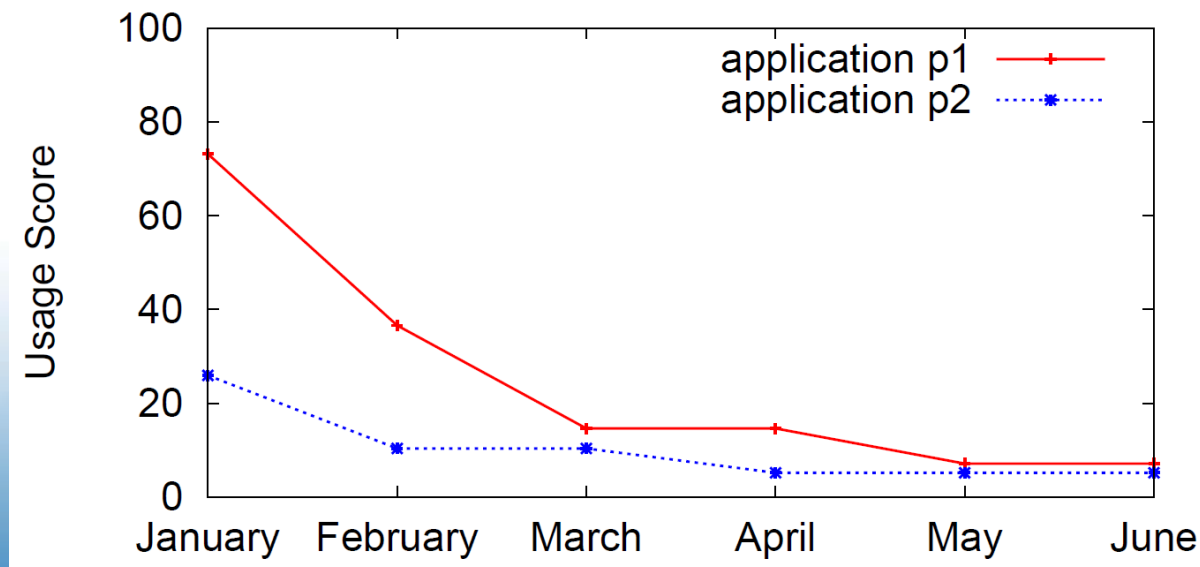
# How to Describe User Preference

- Passively observe how the apps are being used
  - The more an app being used the more the user likes it
  - The usage patterns of the different apps thus can be considered as an objective reflection of the user's taste
- Usage Score
  - **R**ecency, **F**requency and **D**uration
  - The weight shows the importance

$$v_{u+p} = w_R v_R + w_F v_F + w_D v_D$$

# Adaptiveness

- Recency is **adaptive** to the changes of usage
- Give frequency and duration a **penalty** according to recency



# Slope One Prediction

- Similarity

$$dev_{j,i} = \sum_{w \in S_{j,i}} \frac{v_{wt-j} - v_{wt-i}}{card(S_{j,i})}$$

	Item 1	Item 2
Mike	2	unrated
Richard	3	7
Stephen	4	6

- Predict the usage score reflecting how the user like an app

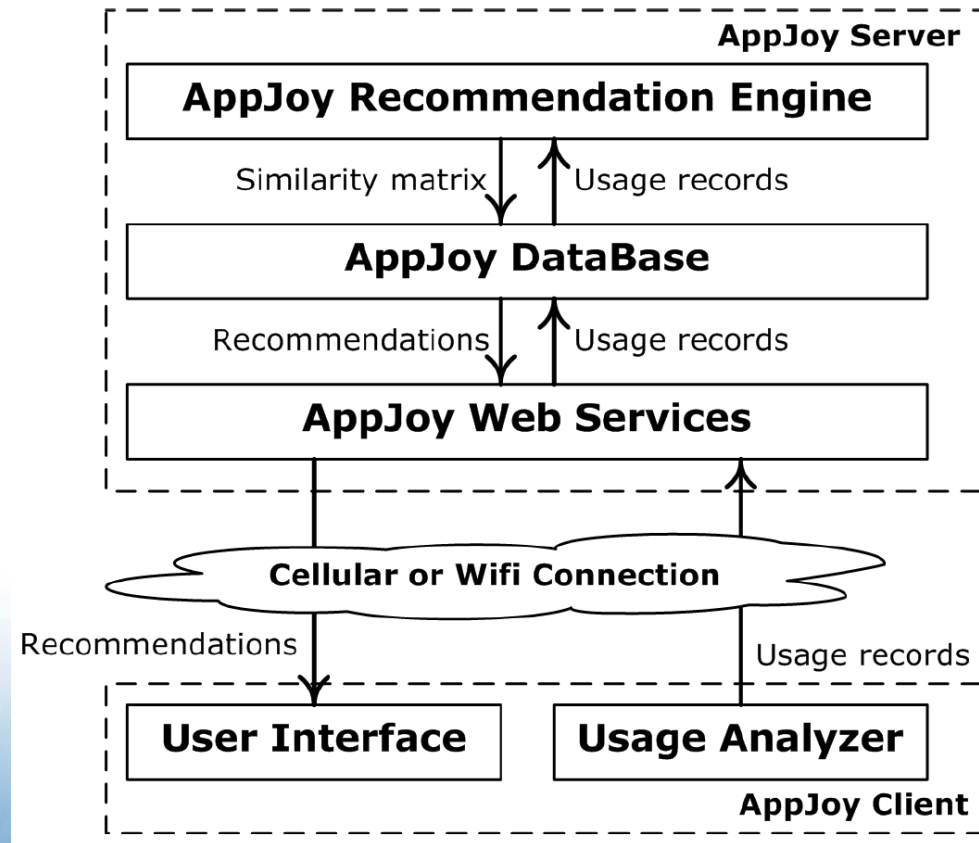
$$P_w(u_j) = \frac{\sum_{i \in R_{u,j}} (dev_{j,i} + u_i) card(S_{j,i})}{\sum_{i \in R_{u,j}} card(S_{j,i})}$$

# Talk Outline

- Recommendation
- Implementation
- Evaluation Results
- Conclusion and Future Works



# Challenges



- How to measure app usage since there is no API provided by Android SDK
- Usability Considerations
- How to identify users

# Monitoring App Usage

- No API to acquire app usage
  - Check every one second which app's activity is in the foreground
- Services are always running
  - Focus on interaction time
- Run a service in the background to do this monitoring

# Usability Considerations

- Slow to load
  - Provide a sense of immediacy
  - Retrieve the list of recommended apps in a background thread
  - Download the icons of apps using a thread pool
  - The perceived wait time is shorten
- Difficult to read and use
  - Bigger fonts and larger areas

# Cookie-based User Auth

- Anonymously usage records
  - Need to identify usage records for personalized recommendation
- Cookie-based Authentication
  - Identify users by device
  - Merge the device identifier and the server token into Cookie

# Talk Outline

- Recommendation
- Implementation
- Evaluation Results
- Conclusion and Future Works

# AppJoy Characteristics

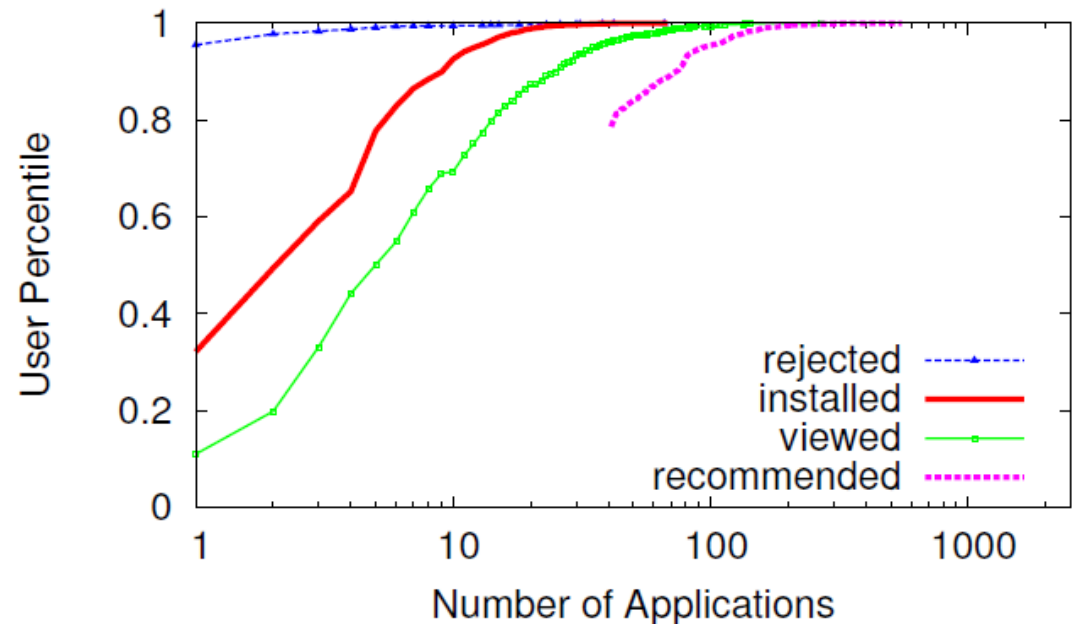
- Release since February 2010
- More than 4,600 users from 10,190 cities in 99 countries
- More than 100 types of smartphones
- 50% of users stayed with AppJoy 10+ days
- A relatively stable organic growth without any advertisement

# Usage Report

- Installed apps, from 3 to 910 (61)
  - frequently-used apps, from 1 to 73 (8)
- 40% of users installed less than 18 of the total 42 categories
  - Exploratory users installed more apps in each category
- 753 users used AppJoy for 30+ days
  - 50% of apps are installed for 11 days
  - 27% of apps are installed for 30 days

# Recommendation Effectiveness

- 4.96% of all recommended apps are installed
  - 7.39% of users installed more than 10

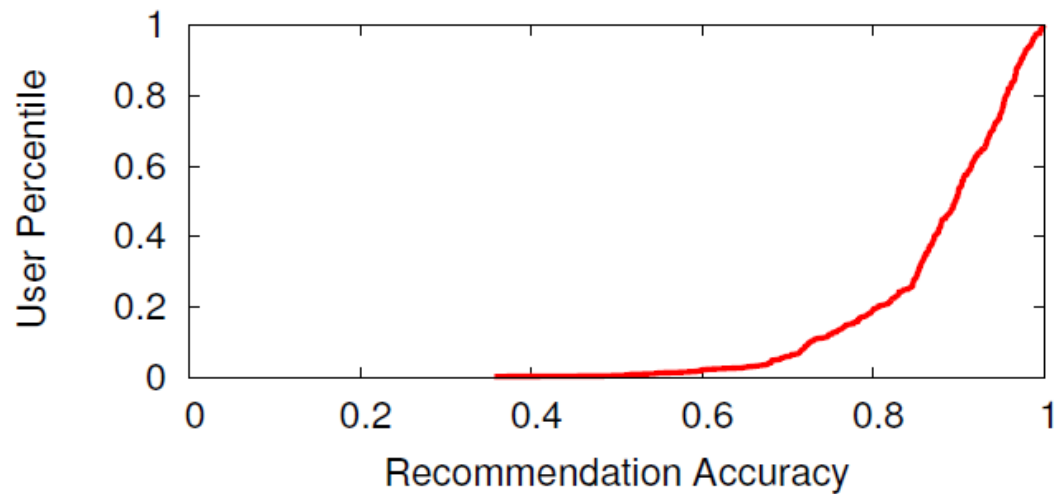


View	24.2%
Rejected	3.57%
Installed	20.5%



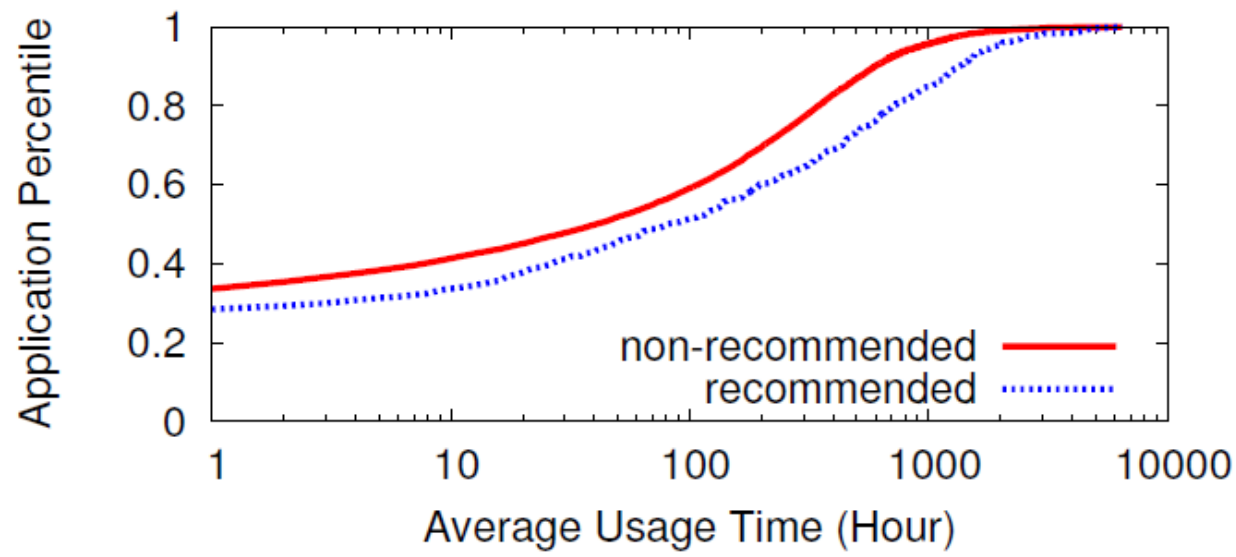
# Recommendation Accuracy

- Can be improved, RMSE = 0.9749
  - Netflix Cinematch, 0.9514
  - Bellkor's Prognatic Chaos, 0.8554
- However, more than 80% accuracy for more than 80% of the users



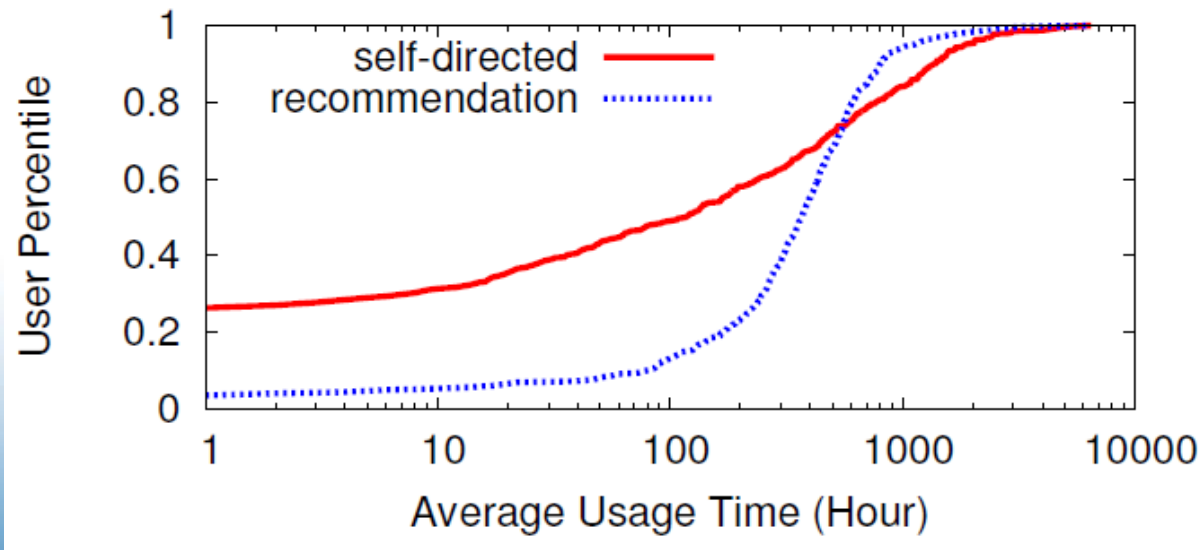
# Recommendations is More Popular

- 2,603 users (v2 and v3)
  - 597 apps installed through AppJoy
  - 14,330 apps not installed through AppJoy



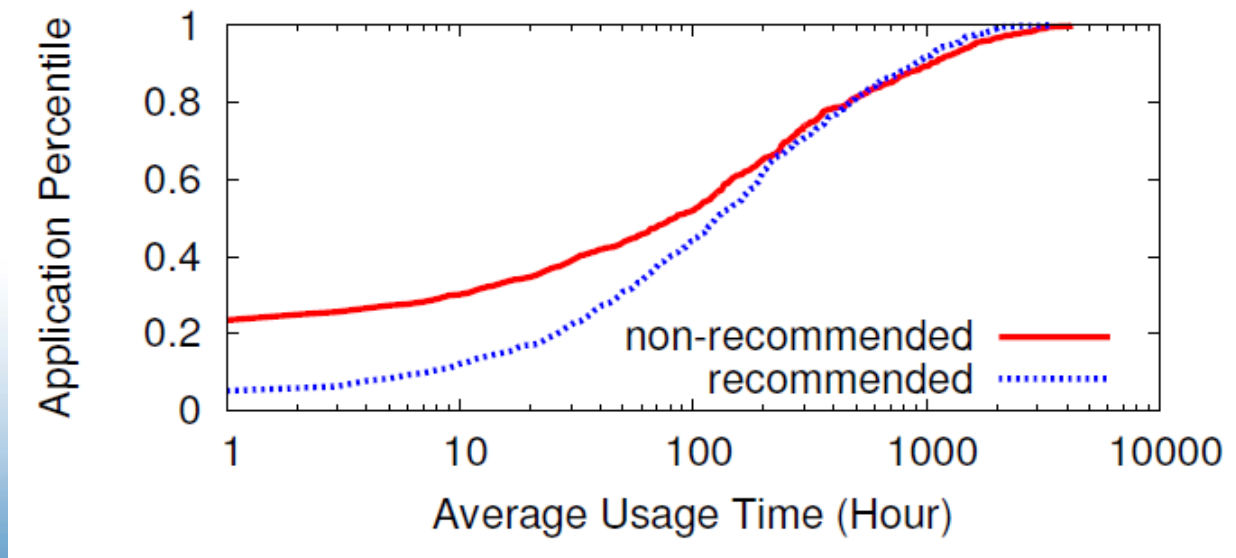
# Meeting Users' Needs

- 597 recommended apps
  - 839 users installed them through AppJoy
  - 1496 users installed them not through AppJoy



# More Interaction Time

- 839 users who installed recommended apps through AppJoy
  - interacted more with recommended apps



# Talk Outline

- Recommendation
- Implementation
- Evaluation Results
- Conclusion and Future Works

# Discussion

- iPhone or Windows Mobile
- Context-aware recommendation
- The little-changed recommendations from relatively stable usage pattern
- Usage record filter against malicious attackers with huge faked usage patterns

# Conclusion

- AppJoy
  - Use collaborative filtering to make personalized mobile application recommendation based on the user's actual usage pattern
  - Completely automatic without requiring manual input
  - Adaptive to the potential changes of the user's application taste
  - Accurate by consuming low battery

# Future Work

- Usability and user study
- Improve recommendation algorithm
  - Integrate the user context
- Perform detailed analysis of app usage pattern at a much larger scale
- Promotion



# Questions and Answers

<http://appjoy.cs.uml.edu>