WTS 2018
April 18-20, Phoenix, Arizona, USA
Wireless Telecommunications Symposium

Predictive Self-Learning Content Recommendation System

Ibrahim Hussein Mwinyi

Husnu Saner Narman

Kuo-chi Fang

Wook-Sung Yoo

Department of Computer Science

Marshall University, USA

narman@marshall.edu

http://mupfc.marshall.edu/~narman/

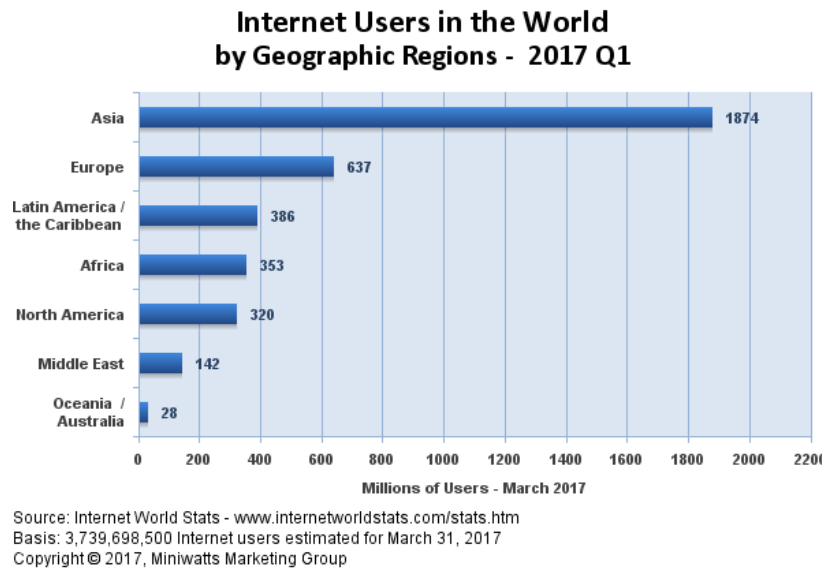
April 2018

Outlines

- Introduction
- Objective
- Proposed Model
- Analysis
- Conclusion

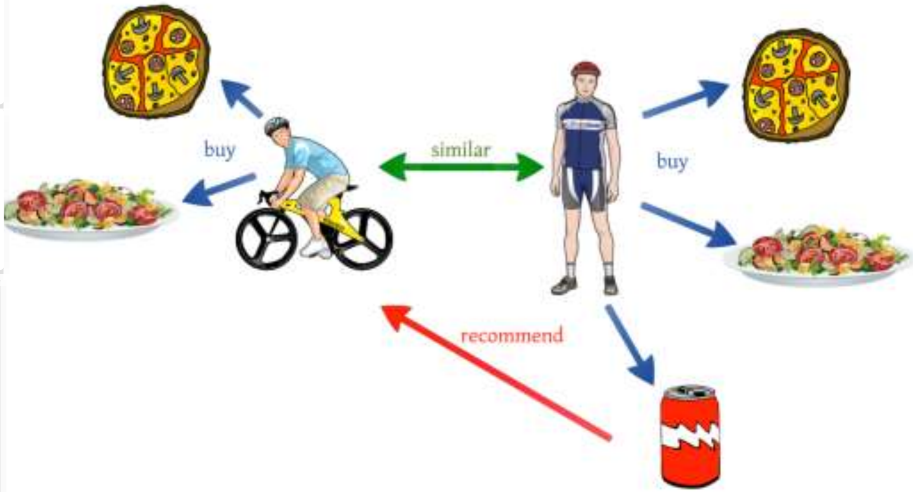
Introduction

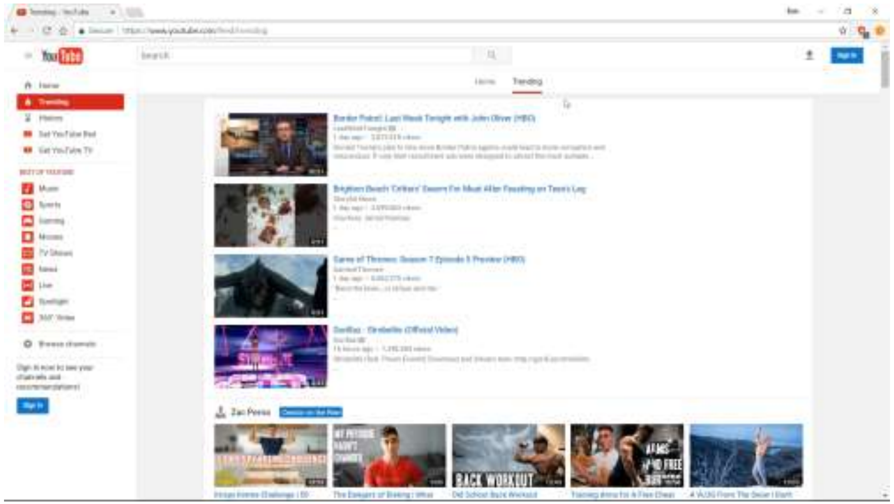
- Millions of daily internet users
- Entertainment, Education, Shopping etc.
- Key feature of online software is recommendation system
- Well known sites:
    - Youtube
    - Netflix
    - Amazon



Recommendation System

- Suggestions





Recommendation System Metric

- Parameters
  - Related content
  - Popularity
  - Channels
  - Location
  - Past Activity
  - Language
  - User Profile

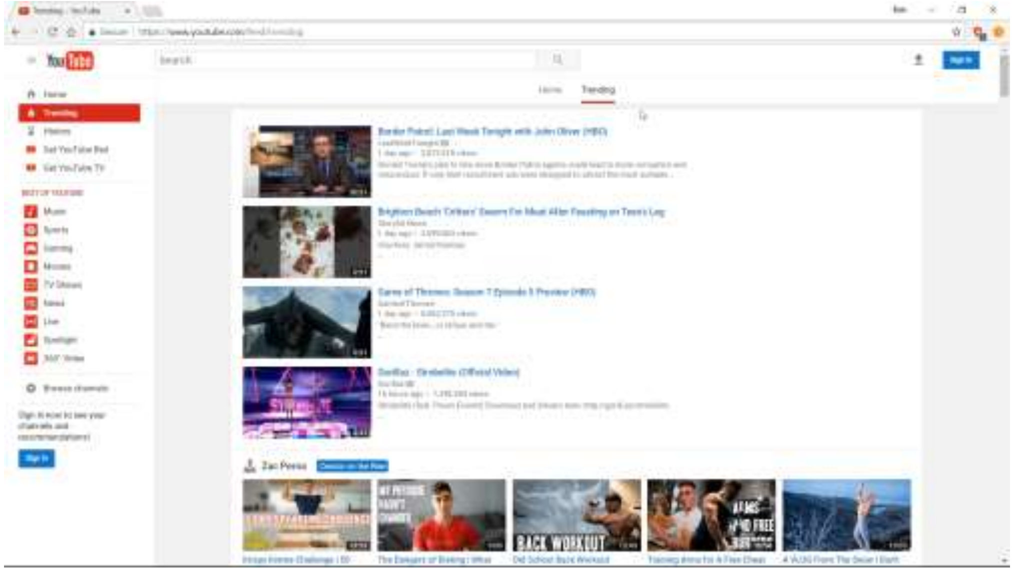
User Input

- Background information
  - Age
  - Location
  - Nationality
  - Occupation
  - Etc…

Ranking

- The most watched
- The most liked

- Input Analysis

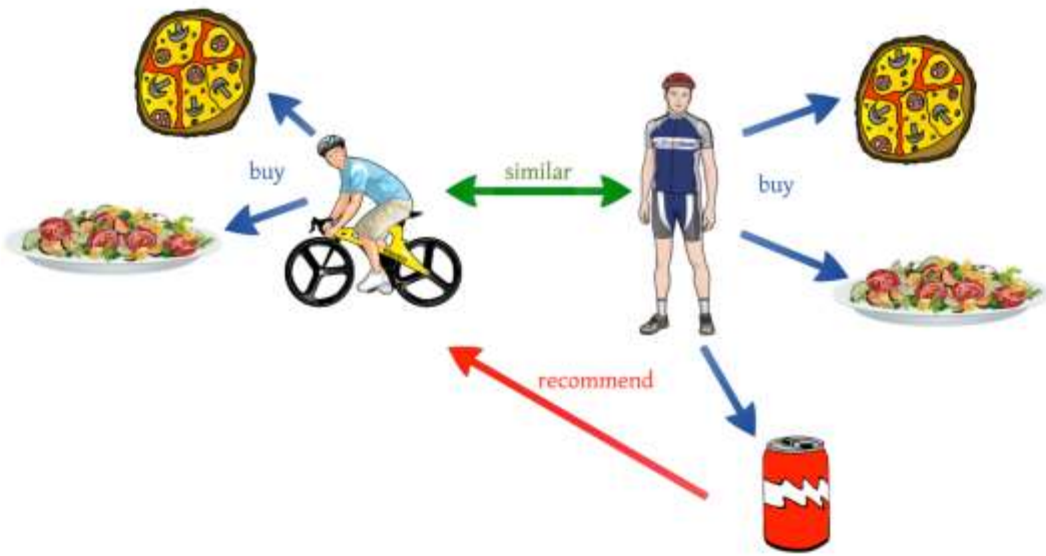


Feedback

- Accuracy
- Safety

Similarity and Grouping

- User history on products will allow the system to produce more suggestions.



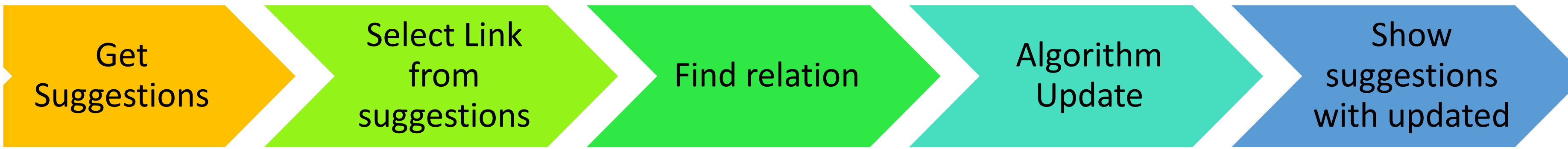
Problem

- Many parameters to consider
- Which parameters are more important for which user

Objective

Propose a predictive self-learning recommendation system that using history information and analyze the user behaviors for the future activity. Then, use a behavior analyzer to update the prediction system by monitoring users selections from suggested contents.

System Flow Model



System Metrics

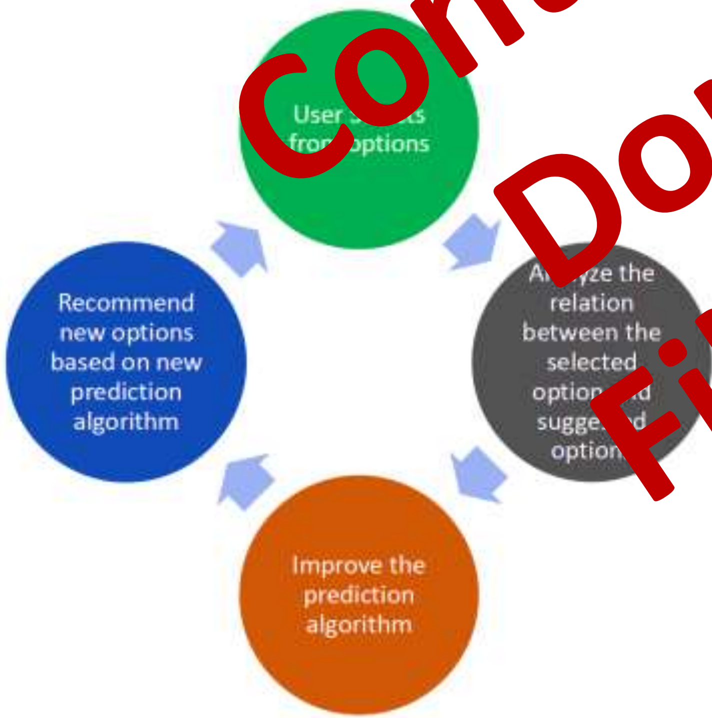
$$R = \alpha Uc + \beta(Fp - Fn) + \left(\gamma \frac{Pl}{10} + \eta \frac{Po}{10} + \kappa \frac{Pn}{10} + \varsigma \frac{pi}{10}\right) + \psi Oc$$

$$\min_{X^1,\ldots,Xnm} \frac{1}{2} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} \left(\left(\theta^{(j)}\right)^T x^{(i)} - y^{(i,j)}\right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^{n} \left({x_k}^{(i)}\right)^2$$

Parameters	
Uc	User Clicks
Fp and Fn	Positive and negative feedback
Pl	Location
Po	Occupation
Pn	Nationality
pi	Interests
Oc	Overall clicks

System Cycling

Continuous Double Filtering



Algorithm 1 Algorithm to improve recommendation systems

Require:

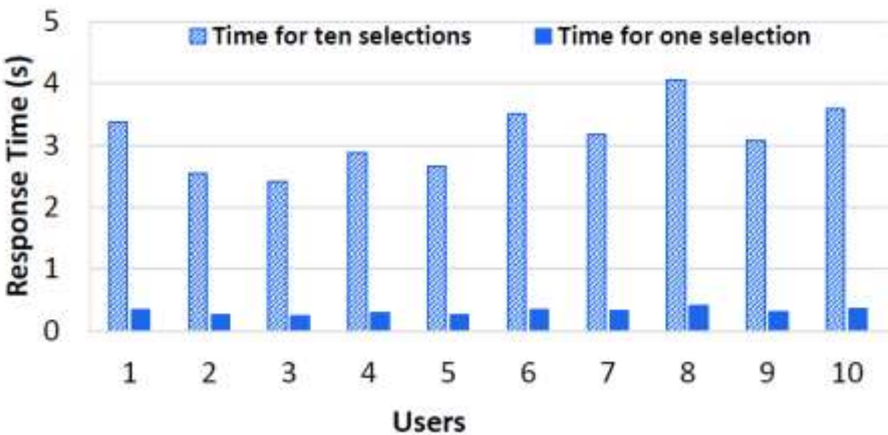
    input - user profiles and past activities
    output - suggestions

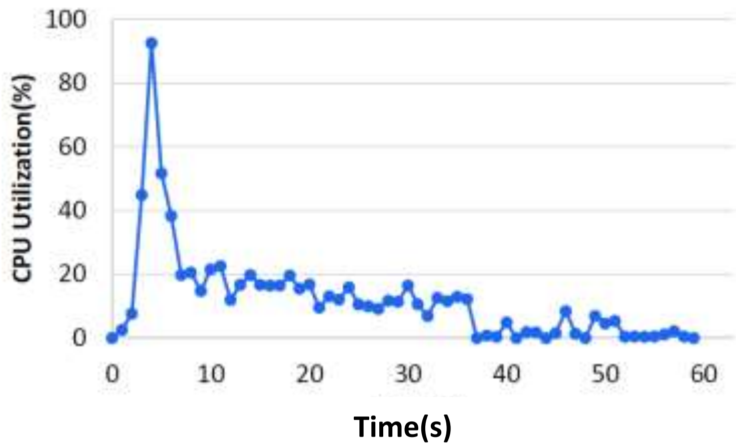
procedure
    while user not done **do**
        display suggestions
        get user selections
        if selection in DB **then**
            display the content
            create a relation: selected and suggested options
            update user recommendation system
            display some suggestions according to the new recommendation and some based on the previous recommendation system
        end if
    end while
end procedure

Simulation-based Analysis

- Data Type – Meta Data from YouTube
- Test Cases
  - Response Time
  - CPU Utilization
  - Suggestion Uniqueness
- Simulation Design
  - Java Implementation
  - Quad Core 2.4GHz, 16GB Memory, 32MB cache

Results

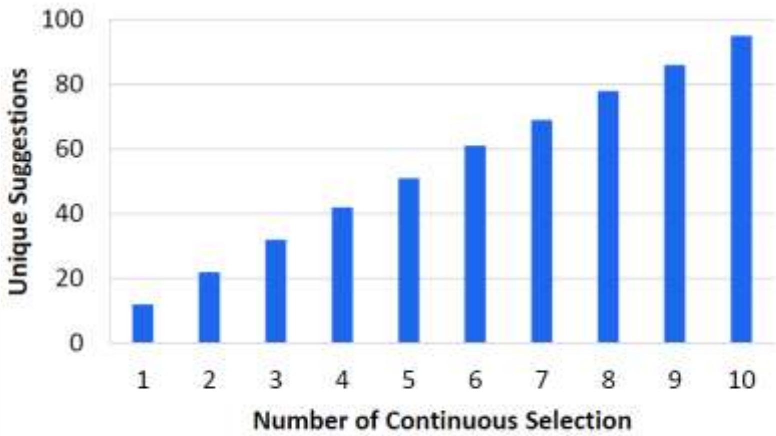


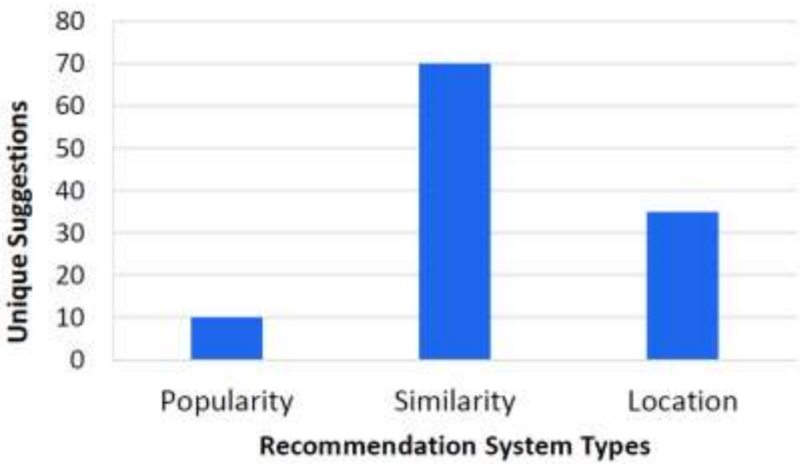


- User selection response time for:
  - One selection
  - Ten selections

- CPU Utilization over 60 seconds

Results





- Unique results returned by the system after continuous user selections over 10 rounds.

- Comparison of unique suggestions return by different recommendation systems

MARSHALL

Conclusion and Future Works

- Predictive Self-Learning Recommendation System.
- Analyzed according to integrity, CPU performance and Time efficiency.
- Need more experiment.
- Creating relation between medias should be investigated more.

Thank You

narman@marshall.edu

http://mupfc.marshall.edu/~narman/