Google News Personalization: Scalable Online Collaborative Filtering

Abhinandan Das, Mayur Datar, Ashutosh Garg, Shyam Rajaram

Google Inc, University of Illinois at Urbana

Paper Review

By

Archana Bhattarai

Introduction to Data Mining

Outline

- Background
- Introduction
- Motivation
- Method
  - System
  - Algorithms
- Result
- Conclusion

Paper: Introduction

- As the topic suggests, this paper talks about a special case of a "Recommender System" specific to Google News scenario for generating personalized recommendations for users of Google News.
- The basic research problem that is addressed by this paper is the challenge of matching the right content to the right user.
- Based on user profile, the system recommends top K stories that user might be interested in.

Background

- Information overflow with the advent of technologies like Internet.
- People are drowning in data pool without getting right information they want.

- Challenge:
  - To find right information.

- Right Information:
  - Something that will answer users' query.
  - Something that user would love to read, listen or see.

- Solution:
  - Search Engines
    - Solve the first requirement
  - ***What if user does not know what to look for ?***

Introduction: Collaborative Filtering

- It is a technology that aims to learn user preferences and make recommendations based on user and community data.

- Example:
  - Amazon: User's past shopping history is used to make recommendations for new products.
  - Netflix, movie recommender
  - Recommendations for clubs, cosmetics, travel locations.
  - Personalized Google News

Motivation

- Google News is visited by several millions in a period of few days.
- There are lots of articles being created each day.
- Scalability is a big issue for such personalized system.
- Moreover, since it is a news based system, the items cannot be static as the articles are changing very fast.

- Existing recommender system thus unsuitable for such need.
- Need for a novel scalable algorithm.

Google News System

- Google news will record the search queries and clicks on news stories.
- Makes previously read articles easily accessible.
- Recommends top stories based on past click history.

- Recommendations based on:
  - Click history.
  - Click history of the community.

- User's click on an article is treated as positive vote.
  - Could be noisy
  - No negative votes

Problem statement

- Given a click history of N users,
  - $U = \{u_{1,} u_{2,} u_{3,} u_{4,} u_{5} \ldots\ldots\ldots\ldots, u_N\}$
- And M items
  - $S = \{s_{1,} s_{2,} \ldots\ldots\ldots\ldots, s_M\}$
- User u with click history set C_u consisting of stories
  - $\{s_{i1,} s_{i2,} \ldots\ldots\ldots\ldots, s_{Cu}\}$

- System is to recommend K stories that user might be interested in.
- Incorporate user feedback instantly.

Related Work :Architectures and algorithm

- Algorithms
  - Memory-based algorithms
    - Predictions made based on past ratings of the user.
    - Weighted average of ratings given by other users
    - Weight is the similarity of users (Pearson correlation coefficient, cosine similarity)
  - Model-based algorithms
    - A model of the user developed based on their past ratings.
    - Use the models to predict unseen items.(Bayesian, clustering etc)

Proposed System

- Mixture of
  - Model based algorithms
    - Probabilistic Latent Semantic Indexing
    - MinHash
  - Memory based algorithms
    - Item co-visitation

- The scores given by each algorithm is combined as
  - $\sum w_a r_s$ where w_a is the weight given to algorithm 'a' and r_s is its rank.

Algorithms

- MinHash
  - A probabilistic clustering method that assigns a pair of users to the same cluster with probability proportional to the overlap between the set of items that these users have voted for.
  - User U is represented by a set of items that she has clicked, $C_{u.}$
  - The similarity between their item-sets is given be :

$$S(u_i, u_j) = \frac{| C_{ui} \cap C_{uj} |}{| C_{ui} \cup C_{uj} |} \qquad \text{(Jaccard Coeffient)}$$

  - Similarity of a user with all other users can be calculated.
    - Not scalable in real time

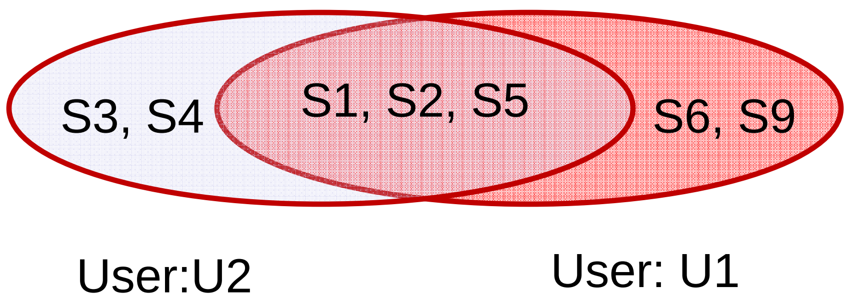
MinHash: Example

- User u1 clicks on the items:

  S1, S2, S5, S6, S9

- Similarly, user u2 clicks on the items:

  S1, S2, S3, S4, S5



- Jaccard Coefficient : 3/7

Algorithms

- Min-Hashing
  - Each hash bucket corresponds to a cluster, that puts two users together in the same cluster with probability equal to their item-set overlap similarity S(u_i, u_j).
  - Randomly permute a set of items(S) and for each user u_u, compute its hash value h(u) as the index of the first item under the permutation that belongs to the user's item set C_u
  - For a random permutation, chosen uniformly over the set of all permutations over S, the probability of two users having same hash value is Jaccard coefficient.

- MapReduce is used for MinHash clustering over large clusters of machines.
- MapReduce is a simple model of computation over large clusters of machines.

Algorithms

- Probabilistic Latent Semantic Indexing[PLSI]
  - With users U and items S, the relationship between users and items is learned by modeling the joint distribution of users and items as a mixture distribution.
  - A hidden variable Z is introduced to capture this relationship, which can be thought of as representing user communities(like minded users) and item communities(like items)
  - Mathematically,

$$P(s/u) = \sum_{z=1}^{L} \underbrace{p(z/u)}_{\text{like users}} \quad \underbrace{p(s/z)}_{\text{like items}}$$

  - The conditional probabilities p(z/u) and p(s/z) are learned from the training data using Expectation maximization algorithm.

PLSI: Concept

User/ News	S1	S2	S3	S4	S5	S6
U1	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}
U2	C_{21}	C_{22}	C_{23}	C_{24}	C_{25}	C_{26}
U3	C_{31}	C_{32}	C_{33}	C_{34}	C_{35}	C_{36}

- Decompose Matrix as, $C = UZS$
- New term 'Z' is introduced.
- Matrix decomposed using Singular Value decomposition

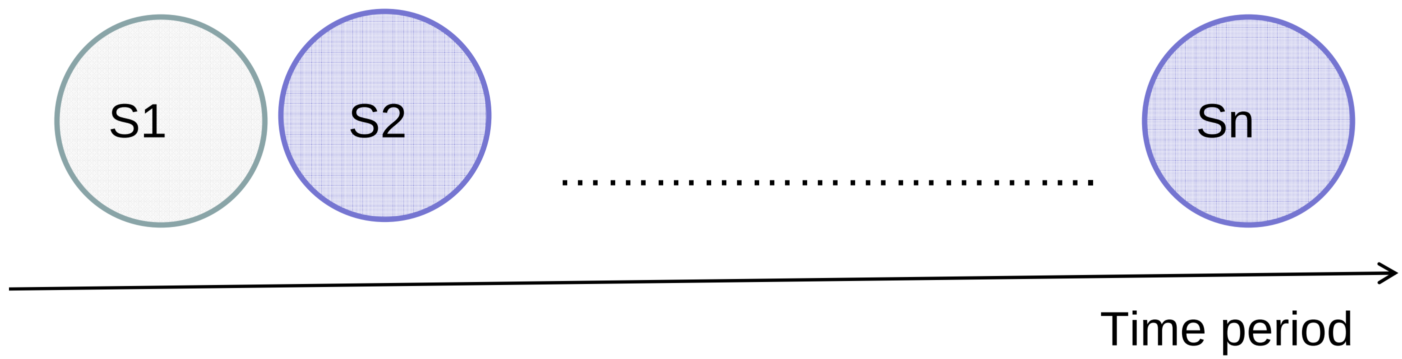
User/ News						
U1	..	..	..	..	...	...
U2	..	..	..	..	..	..
U3	..	..	..	..	..	..

Z
is a
diagonal
matrix

S1	…	…..	….
S2	….	….	…
S3	…	…..	….
S4	…..	…..	…..
S5	…..	…..	…
S6	…	…	….

Algorithms

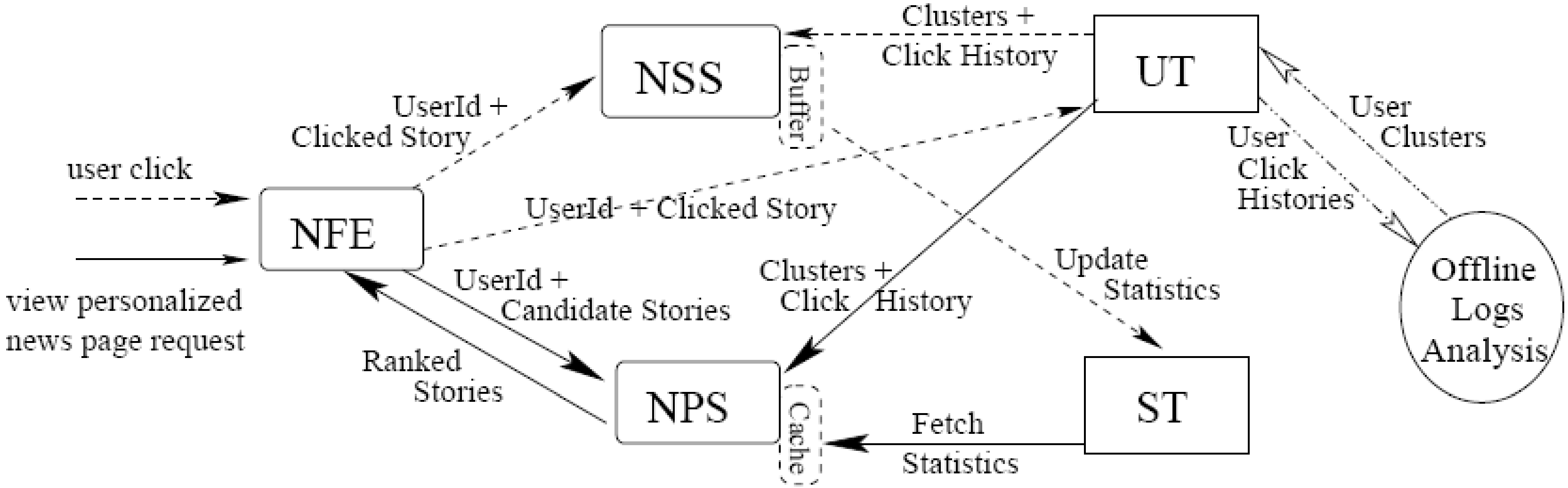
- Co-visitation
  - An event in which two stories are clicked by the same user within a certain time interval.
  - For a user u, covisitation based recommendation score is generated for a candidate item s
  - For every item s_i in the user's click history, a lookup for the entry pair si, s is gotten.
  - The value stored in the entry is added and then normalized by the sum of all entries for s_i.



Data stored

- User Table:
  - Cluster information (MinHash and PLSI)
  - Click history
- Story Table:
  - Cluster Statistics: How many times was the story S clicked on by users from each cluster C.
  - Co-visitation: How many times was story S co-visited with each story S’

System Components



NFE: News Front End
NSS: News Statistics Server
NPS: News Personalization Server
UT: User Table
ST: Story Table

Evaluation Results

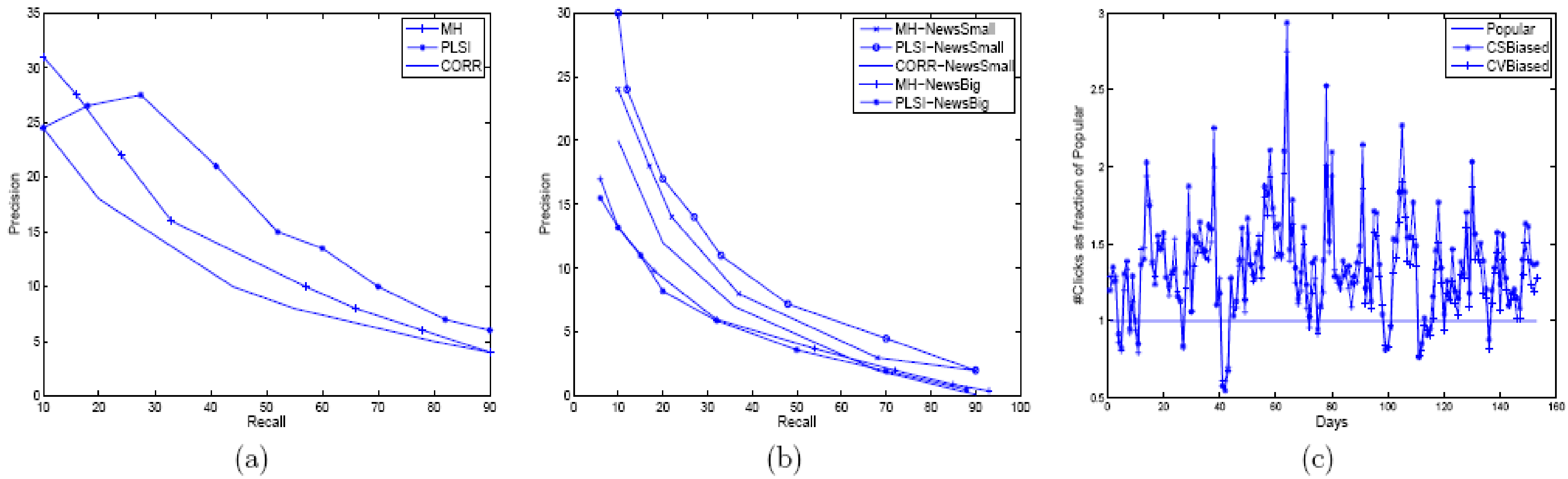


Figure 3: (a) Precision recall curves for the MovieLens dataset. (b) Precision recall curves for the GoogleNews dataset. (c) Live traffic click ratios for different algorithms with baseline as Popular algorithm.

Conclusion and Future Work

- Algorithms for scalable real time recommendation engines presented.
- Presented a novel approach to cluster dynamic datasets using MinHash and PLSI.
- Scalability and quality have a tradeoff.
- The system is content independent and thus easily extendible to other domains.

- As a future work, suitable algorithm can be explored to determine how to combine scores from different algorithms.

Analysis

- The paper has successfully addressed the problem of scalability for large recommender systems.
- It has only looked at the content independent features of articles.
- Thus the content dependent features are out of scope for the paper.
- Evaluation based on content could be an open research problem.
- It can be argued that instead of only considering user click for clustering similar users, content based clustering of the stories could open up more similarity metrics for the recommendation system.
- The precision lies around 30% for the current system showing that more study needs to be done in the field.

Thank You!!!
Any questions ?

Hint: use coalesce instead of collect