RecDB in Action - Recommendation Made Easy In Relational Databases

Mohamed Sarwat  James Avery  Mohamed F. Mokbel
University of Minnesota, Twin Cities

Motivation

(1) Online Recommendation Processing

- We live in an increasingly social and “real-time” world:
  - Number of things to recommend is growing exponentially.
  - Users expressing opinions faster than ever (b) Recommendations change second-by-second

- Offline processing Drawbacks:
  - Less accuracy since the model is maintained periodically and hence does not return real-time fresh recommendations.
  - Not efficient for on-demand filtered recommendation
  - Incurs lots of storage overhead.

(2) Recommendation Filtering and Integration with Data

Recommend User A Five Movies
Recommend User A Five Comedy Movies
Requires joining recommendation with movies to only retrieve comedy movies
Builds the recommendation model entries only for Comedy Movies

RecDB Architecture & Features

Existing Recommender System Architectures

- External Libraries: (1) generate recommendations offline and hence (2) No efficient way for on-demand filtered recommendation
- DBMS User-Defined-Functions: (1) provide online recommendation, but (2) do not leverage the full power of the query optimizer

MyCima – A Movie Recommendation App

SindbadRec – A Restaurant Recommendation App

SciVisor – A Paper Recommendation App

Recommender Queries

Q1: Recommend 5 movies to user 1 using item-based CF algorithm
SELECT A.mid FROM Ratings A
RECOMMEND A.mid TO A.uid
ON A.rating USING ItemCosCF
WHERE A.uid = 1
ORDER BY A.rating LIMIT 5

Q2: Recommend 5 Comedy movies to user 1 using ItemCosCF algorithm
SELECT A.mid FROM Ratings A, Movies M
RECOMMEND A.mid TO A.uid
ON A.rating USING ItemCosCF
WHERE A.uid = 1 AND A.mid = E.mid
AND E.genre = 'Comedy'
ORDER BY A.rating LIMIT 5

Q3: Recommend 5 restaurants to user 1 using SVD algorithm
SELECT A.pid FROM CheckIns A
RECOMMEND A.pid TO A.uid
ON A.visited USING SVD
WHERE A.uid = 1
ORDER BY A.visited LIMIT 5

Q4: Recommend 5 restaurants in “Riva Del Garda” to user 1 using SVD algorithm
SELECT A.pid FROM CheckIns A, Restaurant B
RECOMMEND A.pid TO A.uid
ON A.visited USING SVD
WHERE A.uid = 1 AND B.rid = C.rid
AND C.location = 'Riva Del Garda'
ORDER BY A.visited LIMIT 5

Q5: Recommend 5 VLDB 2013 Papers to user 1 using UserPearCF algorithm
SELECT A.pid FROM DBLP Citations A, Papers B
RECOMMEND A.pid TO A.uid
ON A.cited USING UserPearCF
WHERE A.uid = 1 AND A.pid = B.pid
AND B.venue = 'VLDB2013'
ORDER BY A.cited LIMIT 5

Q6: Recommend 5 VLDB 2013 Papers, presented on the second day of the conference, to user 1 using UserPearCF algorithm
SELECT B.title, Citation, Citation.Cite FROM DBLP Citations A, Papers B, VLDB2013 Program C
RECOMMEND A.pid TO A.uid
ON A.cited USING UserPearCF
WHERE A.uid = 1 AND A.pid = B.pid AND B.pid = C.pid
AND B.venue = 'VLDB2013' AND C.day = 2
ORDER BY A.cited LIMIT 5

Experimental Results

Varying Data Size (ItemCosCF)
Varying Selectivity (SVD)

This work is supported in part by the National Science Foundation under Grants IIS-0952977 and IIS-1218168