Bayesian-inference Based Recommendation in Online Social Networks

Author: Xiwang Yang, Yang Guo, and Yong Liu

Presentor: Guoliang Liu
Outline

• Introduction
• Related Work
• Motivation
• Bayesian Network Introduction
• Bayesian-inference Based Recommendation System Design
• Distributed Protocol Design
• Recommend Friends
Introduction

• Example: Movie recommendation
• Experiment from Flixster and Epinions
• Related Issues:
  – Measure the rating similarity
  – Construct a Bayesian network and query propagation tree
  – Design iterative algorithm to calculate the most probable recommendation
  – Design distributed protocols to implement
Related Work

• Recommender System
  - Content-based recommendation
  - Collaborative Filtering (CF)-based recommendation

• Trust v.s. conditional probability
Motivation

• Using rating similarity to infer the “most probable” rating for John.
Bayesian Network Introduction and Analysis of one example

\[ \hat{R}_{J|R_A=r_A} \triangleq \arg\max_{r} P(R_J = r | R_A = r_A). \]

\[ \hat{R}_{J|r_A,r_B,r_C} \triangleq \arg\max_{r} P(R_J = r | r_A, r_B, r_C), \]

\[ P(r_A, r_B, r_C | r_J) = P(r_A | r_J)P(r_B | r_J)P(r_C | r_J). \]

**Fig. 1. Recommendation social network**

\[ P(r_J|r_A, r_B, r_C) = \frac{P(r_A, r_B, r_C | r_J)P(r_J)}{\sum_{r} P(r_A, r_B, r_C | R_J = r)P(R_J = r)}. \]
Bayesian-inference Based Recommendation System Design

• Framework
  – Users are connected in a social network $G=(V,E)$
  – Current user has rating, return; otherwise, relay
  – A query will be dropped after pre-defined number of hops
  – Recommendation agent calculates a score for the querying user using a Bayesian network
Bayesian-inference Based Recommendation System Design

• Recommendation Propagation Tree
  – S: query set
  – $\mathcal{L} = \{L_i \in V, i = 1, 2, \ldots, k\}$: responding set
  – M: intermediate

Fig. 2. Recommendation Propagation Tree as a Bayesian network
Bayesian-inference Based Recommendation System Design

• Bayesian Network

\[ Pr(R_S = s | R_{L_i} = r_i, 1 \leq i \leq K) \]

\[ C_m \text{ as the set of its direct children} \]

\[ D_m \text{ as the set of recommenders in the subtree rooted at } m: \]

\[ D_m \triangleq \{L_i \in \mathcal{L} : L_i \text{ is in the subtree rooted at } m\}. \]

We define the probabilistic event that recommender \( L_i \) gives rating \( r_i \) as \( \Phi(L_i) \triangleq \{R_{L_i} = r_i\} \). Then the event of the joint ratings of all recommenders under \( m \) can be composed as

\[ \Psi_m \triangleq \{R_{L_i} = r_i, L_i \in D_m\} = \bigcap_{L_i \in D_m} \Phi(L_i). \]
Bayesian-inference Based Recommendation System Design

• Bayesian Network

Our goal is to estimate $P(R_S = s | \Psi_S)$. Given the rating range of $[1, N]$, following the Bayesian rule, we have

$$P(R_S = s | \Psi_S) = \frac{P(\Psi_S | R_S = s)P(R_S = s)}{\sum_{r=1}^{N} P(\Psi_S | R_S = r)P(R_S = r)}.$$  

Since $D_S = \bigcup_{c \in C_S} D_c$ and $\Psi_S = \bigcap_{c \in C_S} \Psi_c$, we have

$$P(\Psi_S | R_S = s) = P(\bigcap_{c \in C_S} \Psi_c | R_S = s) = \prod_{c \in C_S} P(\Psi_c | R_S = s),$$

where the last equivalence is established by the conditional independence in the Bayesian network. If a child $c$ is a recommender, i.e., $c \in \mathcal{L}$, then $D_c = \{c\}$, and $P(\Psi_c | R_S = s) = P(R_c = r_c | R_S = s)$, which can be directly obtained from the conditional probability between $c$ and its parent $S$. If $c$ is a relay node, i.e. $c \notin \mathcal{L}$, then we have

$$P(\Psi_c | R_S = s) = \sum_{i=1}^{N} P(\Psi_c \cap \{R_c = i\} | R_S = s)$$

$$= \sum_{i=1}^{N} P(\Psi_c | R_c = i)P(R_c = i | R_S = s),$$
Bayesian-inference Based Recommendation System Design

• Recommendation Calculation
  – Most probable recommendation
    \[
    \hat{S}^{MP} \triangleq \operatorname*{argmax}_{1 \leq s \leq N} P(R_S = s | \Psi_S). \tag{4}
    \]
  – Bayes Mean Square Error estimator
    \[
    \hat{S}^{MSE} \triangleq E[S | \Psi_S] = \sum_{s=1}^{N} s P(R_S = s | \Psi_S). \tag{5}
    \]
Distributed Protocol Design

• User-recommendation Interface
• Learning Module
• Recommendation Engine
Coping with code start and rating sparseness

Fig. 3. User Confidence level Input Interface
Recommend Friends