Adaptive Collaborative Topic Modeling for Online Recommendation

Al-Ghossein, Marie, et al. (2018)

Proceedings of the 12th ACM Conference on Recommender Systems

Presenter: Sarah Kim

2018.12.07

1. Introduction

- Because data are generated continuously in real-world, a Recommender System (RS) should be able to handle user-item feedback and the availability of new items in real-time.
- Learning from data streams is essential to account for **concept drifts** which occur when the definition of modeled concepts changes over time.
- This paper propose a online RS in a dynamic environment where
  1. users interact with items in real time;
  2. new items are expected to arrive with textual description.

1. Introduction

- Our approach combines AWILDA and incremental matrix factorization (MF).
  - AWILDA: an adaptive version of online LDA that is able to analyze and model documents arriving in a stream;
  - Incremental MF: a variant of MF adpated to the incremental nature of data streams.
- The proposed method has capacity of automatically detecting and adapting to drifts, hence it suitable for real-world scenarios where changes in topics of document streams are frequently happening.

Matrix Factorization (MF)

- For the user u, the item i, $R_{ui} = 1$ if u interacted with i, and 0 otherwise.
- D: the set of observed interactions
- K: the number of latent factors
- MF minimizes the following objective function:

$$\underset{P,Q}{\text{argmin}} \sum_{(u,i)\in D} \left(R_{ui} - P_u Q_i^\top\right)^2 + \lambda_u \|P_u\|^2 + \lambda_i \|Q_i\|^2$$

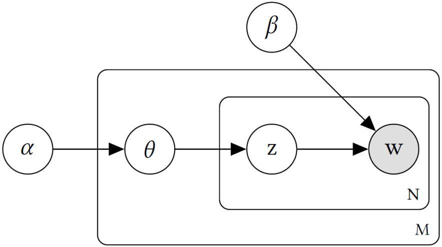
where P_u, Q_i are K-dimensional latent vector for u, i, and λ_u, λ_i are regularization parameters.

Incremental MF

- Observations $< u, i >$ are received one after the other;
- For each received observation, P and Q are updated using the gradient of the objective for this observations only.
- If an user or an item are observed for the first time, they are added to P or Q with random initialization, and the values of P and Q are then updated using the observation.

- Latent Dirichlet Allocation (LDA) is a generative model describing text documents and corpora.
- In LDA, a document is described as a mixture of topics, and a topic corresponds to a word distribution.
- For training of LDA, use either Gibbs sampling or variational inference.
- Online LDA is an online version of variational inference based on online stochastic optimization.

- LDA assumes the following generative process for a corpus consisting of M documents each of length N_i:
  1. Choose $\theta_i \sim \mathrm{Dir}(\alpha)$, where $i = 1, \ldots, M$,
  2. Choose $\phi_k \sim \mathrm{Dir}(\beta)$, where $k = 1, \ldots, K$ and K is the number of topics,
  3. For each of the word positions i, j, where $i = 1, \ldots, M$ and $j = 1, \ldots, N_i$,
     - (a) Choose a topic $z_{i,j} \sim \mathrm{Multinomial}(\theta_i)$.
     - (b) Choose a word $w_{i,j} \sim \mathrm{Multinomial}(\phi_{z_{i,j}})$.



- Adaptive Sliding Window algorithm (ADWIN) uses a sliding window W to detect a change in a sereis of one-dimensional observations.
- A drift is detected if W can be separate into two subwindows $W = W_0 W_1$ s.t. the difference of means μ_{W_0} and μ_{W_1} is large enough.

- **Adaptive Window based Incremental LDA (AWILDA)** used for topic drift detection in LDA which is a expansion of ADWIN.
- AWILDA is based on two models of LDA:
  - LDA_m is used for document modeling
  - LDA_d is used for the detection of drifts only
- AWILDA works as follows: When a new document is received,
  1. Compute likelihood $\mathcal{L} = p(w|LDA_d)$.
  2. Process $\mathcal{L}$ with ADWIN for detecting a drift.
  3. If ADWIN detects a drift for window decomposition $W = W_0 W_1$: Retrain LDA_m, LDA_d based on the documents in W_1.
  4. Update LDA_m from the new document based on the online LDA algorithm.

3. Adaptive Collaborative Topic Modeling

3.2. CoAWILDA

Algorithm 1 Overview of CoAWILDA

Data: set of observations O

Input: number of latent factors K, learning rate η, regularization parameters λ_u and λ_i

Output: P, Q

for o in O **do**
    if $o = \langle i, doc_i \rangle$ **then** ▷ new item added
        $\theta_i \leftarrow AWILDA(doc_i)$
        $\epsilon_i \sim \mathcal{N}(0,\ \lambda_i^{-1} I_K)$
        $Q_i \leftarrow \theta_i + \epsilon_i$
    end if
    if $o = \langle u, i \rangle$ **then** ▷ interaction received
        if $u \notin Rows(P)$ **then** ▷ new user observed
            $P_u \sim \mathcal{N}(0,\ \lambda_u^{-1} I_K)$
        end if
        $e_{ui} \leftarrow 1 - P_u . Q_i^T$
        $P_u \leftarrow P_u + \eta(e_{ui} Q_i - \lambda_u P_u)$
        $\epsilon_i \leftarrow \epsilon_i + \eta(e_{ui} P_u - \lambda_i \epsilon_i)$
        $Q_i \leftarrow \theta_i + \epsilon_i$
    end if
end for

---

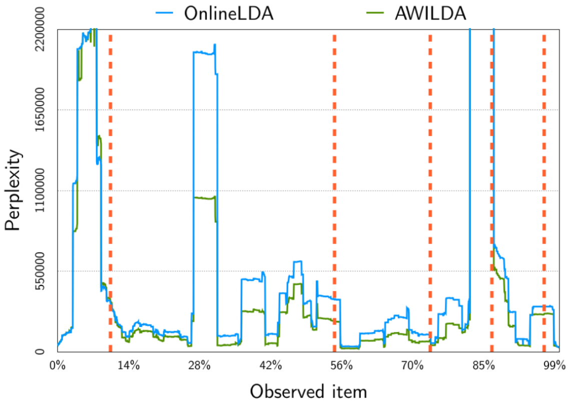
4. Experiments

- Dataset: The *plista* dataset contains a collection of news articles published in German on several news portals (32,706,307 interactions from 1,362,097 users on 8,318 news articles).
- Evaluation measure for topic modeling:

$$perplexity(D_{test}) = \exp\left\{-\frac{\sum_{d=1}^{M} \log_2 p(w_d)}{\sum_{d=1}^{M} N_d}\right\},$$

  where D_{test} is unseen documents, M is the number of documents in D_{test}.
- Evaluation measure for online recommendation: recall@N

- Performance evaluation of online LDA and AWILDA on *plista*



4. Experiments

4.2. Performance of CoAWILDA for online recommendation

- Performance evaluation using recall@N on *plista*