Bayesian recommendation engine on a large streaming sparse data

Jian Kang, Fei Wang

Abstract

The abstract goes here.

Index Terms

IEEE, IEEEtran, journal, LATEX, paper, template.

I. Introduction

II. Setup

Suppose there are n users and m items, respectively indexed by $i = 1, \dots, n$ and $j = 1, \dots, m$. For user i and item j, we observe his/her rating or response value Y_{ij} on item j, which could be a binary , categorical, or continuous variable, and $W_i \in \mathbb{R}^q$ and $X_j \in \mathbb{R}^p$ for a user feature and an item feature, respectively. Concretely, we represent all Y_{ij} by an $n \times m$ matrix $Y \in \mathbb{R}^{n \times m}$, and use $W \in \mathbb{R}^{p \times n}$ to denote $(W_1, \dots, W_n)$ and $X \in \mathbb{R}^{q \times m}$ for $(X_1, \dots, X_m)$. In the literature of recommender system, Y_{ij}, sometimes, is classified into *explicit feedback* and *implicit feedback* [1]. For instance, star ratings for movies and hitting thumbs-up/down buttons are explicit feedback, but purchase history, browsing history, and search patterns are implicit feedback. From the perspective of modeling, however, there is no essential difference between the explicit feedback and implicit one. Similarly, the covariates W and X are treated as contextual information in recommender system that could impact the user response matrix Y.

It is commonly known that the user response matrix Y is sparse, reflecting that customers only have chance to try a few items so that most of Y_{ij} are missing. However, to keep the possibility that any item can be recommended to a specific user in a recommendation engine, we assume in Assumption 1 that the probability that user i selects item j is bounded below by a small positive constant.

Assumption 1. *Let $R_{ij} = 1$ denote that Y_{ij} is observed and $R_{ij} = 0$ otherwise when Y_{ij} is missing. We assume there exists $\delta > 0$ such that*

$$\min_{1 \le i \le n} \min_{1 \le j \le m} P(R_{ij} = 1) > \delta > 0.$$

This assumption is similar to the random zero assumption, instead of structural zeros, considered in the research of multivariate discrete data [2]. Under this assumption we assure that customers eventually will try all items as time goes on although the associated likelihood might be extremely small.

A. Cold-start, streaming data and user feedback

Arranging all missing indicators R_{ij} into a matrix $R \in \mathbb{R}^{n \times m}$ and defining its row partition and column partition

$$R = \begin{pmatrix} R_{11} & R_{12} & \dots & R_{1m} \\ R_{21} & R_{22} & \dots & R_{2m} \\ \vdots & & \vdots & \\ R_{n1} & R_{n2} & \dots & R_{nm} \end{pmatrix} \triangleq \begin{pmatrix} R_{\cdot 1} & \dots & R_{\cdot m} \end{pmatrix} \triangleq \begin{pmatrix} R_{1\cdot} \\ \vdots \\ R_{n\cdot} \end{pmatrix},$$

we discuss *cold-start problem*, which is common in the application of recommender system [3, 4]. The first type of cold-start problem is the *new user problem*, corresponding to $\|R_{i\cdot}\|_1 = 0$ for some i; the second cold-start problem is the *new item problem* defined by $\|R_{\cdot j}\|_1 = 0$ for some j. When a new user i' is introduced in the data, $\|R_{i'\cdot}\|_1 = 0$ so that we have to rely on the association in user feature vectors W to predict the response from user i'. When a new item j' is added to the system, we know $\|R_{\cdot j'}\|_1 = 0$ and therefore the item j' has to be recommended according to the correlation in the item feature X.

Another issue in real application is that data $\mathcal{D} \triangleq \{Y, W, X, R\}$ is a *streaming data* collected in an online way. If the current data at time t is denoted as $\mathcal{D}^t \triangleq \{Y^t, W^t, X^t, R^t\}$, the data at time $t+1$, $\mathcal{D}^{t+1} \triangleq \{Y^{t+1}, W^{t+1}, X^{t+1}, R^{t+1}\}$, is the union of $\mathcal{D}^t$ and $\tilde{\mathcal{D}}^t \triangleq \{\tilde{Y}^t, \tilde{W}^t, \tilde{X}^t, \tilde{R}^t\}$, where $\tilde{Y}^t \in \mathbb{R}^{n_\delta \times m_\delta}$, $\tilde{W}^t \in \mathbb{R}^{n_\delta \times m_\delta}$, $\tilde{X}^t \in \mathbb{R}^{n_\delta \times p}$, and $\tilde{W}^t \in \mathbb{R}^{m_\delta \times q}$.

Manuscript received April 19, 2005; revised August 26, 2015.

Assumption 2. *At time $t+1$, the data $\mathcal{D}^{t+1}$ is defined as*

$$\mathcal{D}^{t+1} = \mathcal{D}^t \cup \tilde{\mathcal{D}}^t, \tag{1}$$

and $n_\delta \geq n$ and $m_\delta \geq m$.

The time interval between t and $t+1$ could be hour, day, week, or month, depending on how often the recommendation engine is refitted. To simply the notation, we drop the superscript t and use $\mathcal{D}$ to denote the current data and $\tilde{\mathcal{D}}$ the updated data.

To consider the cold-start problem in the streaming data, Assumption 3 assumes that cold-start problem exists only in the updated data.

Assumption 3. *R and $\tilde{R}$ satisfy*

$$\min\left\{\min_{1\leq i\leq n}\|R_{i\cdot}\|_1, \min_{1\leq j\leq m}\|R_{\cdot j}\|_1\right\} > 0, \quad \min\left\{\min_{1\leq i\leq n_\delta}\|\tilde{R}_{i\cdot}\|_1, \min_{1\leq j\leq m_\delta}\|\tilde{R}_{\cdot j}\|_1\right\} = 0.$$

This assumption exactly mimics the scenario when a recommendation engine is implemented. At the development stage, researchers build an off-line engine based on all existing customers and items. Only after the engine is online, new users or items start to show up.

This formulation of current data and update data also facilitates the incorporation of *user feedback* [5], $\tilde{Y}$, into a running engine after the engine is online for a while. If the updated response $\tilde{Y}$ contains users already included in the current response Y, we could modify the current recommendation engine to generate better recommended items. If $\tilde{Y}$ are all from new users, we could apply what have learned from the current Y to predict what is most likely to appeal the new users.

B. A new loss function in recommendation engine

Let J be an random variable denoting items and taking values from $\{1,\ldots,m\}$.

$$\hat{J} = \arg\max_J \ E_{R_i}\Big\{E_{Y_i}\{1_{Y_i\in F} Y_i \mid R_i = 1, J\} \mid J\Big\}. \tag{2}$$

III. Bayesian Hierarchical Model

On the current data $\mathcal{D}^t$, we construct a Bayesian hierarchical model. When $R_{ij}=1$, we assume Y_{ij} given W_i and X_j follows a generalized linear mixed effects model

$$g\{E(Y_{ij} \mid R_{ij}=1)\} = W_i^T\beta_Y + X_j^T\alpha_Y + U_i^T V_j, \tag{3}$$
$$U_i \sim \mathrm{N}(0, \sigma_U^2 I_L), \tag{4}$$
$$V_j \sim \mathrm{N}(0, \sigma_V^2 I_L), \tag{5}$$
$$R_{ij} = I(Z_{ij} > 0), \tag{6}$$
$$Z_{ij} \sim \mathrm{N}(\tilde{U}_i^T\tilde{V}_j + W_i^T\beta_Z + X_j^T\alpha_Z, 1) \tag{7}$$
$$\tilde{U}_i \sim \mathrm{N}(0, \tilde{\sigma}_U^2 I_L), \tag{8}$$
$$\tilde{V}_j \sim \mathrm{N}(0, \tilde{\sigma}_V^2 I_L), \tag{9}$$

where $U_i = (U_{i1},\ldots,U_{iL})^T$ and $V_j = (V_{j1},\ldots,V_{jL})^T$ are L-dimensional random effects that account for the variability not explained by the W and X.

IV. Streaming data and update

Appendix A

Appendix one text goes here.

Acknowledgment

The authors would like to thank...

References

[1] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative filtering for implicit feedback datasets," in *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*. IEEE, 2008, pp. 263–272.

[2] Y. M. M. Bishop, *Discrete Multivariate Analysis: Theory and Practice*. MIT press, 1975.

[3] L. H. Son, "Dealing with the new user cold-start problem in recommender systems: A comparative review," *Information Systems*, vol. 58, pp. 87 – 104, 2016.

[4] M. Volkovs, G. Yu, and T. Poutanen, "Dropoutnet: Addressing cold start in recommender systems," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., 2017, pp. 4957–4966.

[5] A. Sinha, D. F. Gleich, and K. Ramani, "Deconvolving feedback loops in recommender systems," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., 2016, pp. 3243–3251.

PLACE
PHOTO
HERE