recommendation. This is supported by the fact that the number of ratings received from different users can differ widely in real-world deployed recommender systems.

To capture such factors, we generalize the rating dominant response model by including both item features and user features. To keep the model tractable and efficient, we introduce a linear combination of the item features, user features and a constant related to the rating scores and pass it through the logistic function to model the response probability,

\[ \mu_{ijk} = \frac{1}{1 + \exp(-(\delta_k + U_i^T \theta_U + V_j^T \theta_V))}. \]  \tag{21} 

We refer to Eq. (21) as context-aware response model, in which the response probability is on a per-user-item-rating basis. More sophisticated relationship definition can be referred to [28, 29, 31]. The PMF integrated with the context-aware response model is named RAPMF-c. Note that by setting \( \theta_U \) and \( \theta_V \) to zero, we can recover the rating dominant response model in Eq. (13).

The log-likelihood of RAPMF-c is in the same structure as RAPMF-r. We only need to substitute \( \mu_{ijk} \) in Eq. (15) by \( \mu_{ijk} \) defined in Eq. (21). Similarly, the gradients of \( \mathcal{L} \) with respect to \( U_i \) and \( V_j \) are

\[
\frac{\partial \mathcal{L}}{\partial U_i} = \beta \sum_{j=1}^{M} \frac{\sum_{k=1}^{D} t_{Ukj} N(k|U^T V, \sigma^2)}{\sum_{k=1}^{D} \alpha_{kij} N(k|U^T V, \sigma^2)} - \sum_{j=1}^{M} (U_i^T V_j - x_{ij})[r_{ij} = 1] V_j - \lambda_U U_i, \tag{22}
\]

\[
\frac{\partial \mathcal{L}}{\partial V_j} = - \beta \sum_{i=1}^{N} \frac{\sum_{k=1}^{D} t_{Vkj} N(k|U^T V, \sigma^2)}{\sum_{k=1}^{D} \alpha_{kij} N(k|U^T V, \sigma^2)} - \sum_{j=1}^{M} (U_i^T V_j - x_{ij})[r_{ij} = 1] U_i - \lambda_V V_j, \tag{23}
\]

where the \( t_{Ukj} \) and \( t_{Vkj} \) is defined as following

\[
t_{Ukj} = g'(\mu_{kj})(-1)^{[r_{ij}=0]} \theta_U - \alpha_{kij}(U_i^T V_j - k) V_j, \tag{24}
\]

\[
t_{Vkj} = g'(\mu_{kj})(-1)^{[r_{ij}=0]} \theta_V - \alpha_{kij}(U_i^T V_j - k) U_i. \tag{25}
\]

Correspondingly, the gradients of \( \mathcal{L} \) with respect to \( \delta_k \), \( \theta_U \) and \( \theta_V \) are

\[
\frac{\partial \mathcal{L}}{\partial \delta_k} = \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{N(l_i U^T V, \sigma^2) g'(\mu_{kj})(-1)^{[r_{ij}=0]}}{\sum_{k=1}^{D} \alpha_{kij} N(k|U^T V, \sigma^2)} - \lambda_{l} \delta_k, \tag{26}
\]

\[
\frac{\partial \mathcal{L}}{\partial \theta_U} = \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{N(k|U^T V, \sigma^2) g'(\mu_{kj})(-1)^{[r_{ij}=0]} U_i}{\sum_{k=1}^{D} \alpha_{kij} N(k|U^T V, \sigma^2)} - \lambda_{U} \theta_U, \tag{27}
\]

\[
\frac{\partial \mathcal{L}}{\partial \theta_V} = \sum_{i=1}^{N} \sum_{j=1}^{M} \frac{N(k|U^T V, \sigma^2) g'(\mu_{kj})(-1)^{[r_{ij}=0]} V_j}{\sum_{k=1}^{D} \alpha_{kij} N(k|U^T V, \sigma^2)} - \lambda_{V} \theta_V. \tag{28}
\]

To learn RAPMF-c, we adopt the alternatively updating scheme to maximize the log-likelihood, where the updating rules of \( U \) and \( V \) are the same as those in Eq. (19). After updating \( U \) and \( V \), we update \( \delta_k \), \( \theta_U \) and \( \theta_V \) by

\[
\vartheta \leftarrow \vartheta + \eta \frac{\partial \mathcal{L}}{\partial \vartheta},
\]

where \( \vartheta \) is replaced by \( \delta_k \), \( \theta_U \) and \( \theta_V \), respectively.

### 3.6 Complexity and Parallelization

The training complexity of RAPMF, \( O(MN) \), can be quite time consuming compared with the PMF, which is linear in number of observations, \( O(|Q|) \). However, we argue that the time spent on training is worthy since it can boost the model performance. More importantly, the prediction complexity of RAPMF is the same as PMF, \( O(K) \), which can be taken as a constant time given a moderate sized \( K \). Since the training procedure can be performed offline, RAPMF can accommodate the hard response time constraint in real-world deployed recommender systems due to the succinct prediction cost.

In addition, RAPMF can be speedup by parallelization. The intensive computation cost, calculating the gradients, can be decoupled and distributed to a cluster of computers. It is also possible to use online learning to speed up the training process [12].

### 4 Experiments and Results

We conduct empirical evaluation to compare the performance of PMF [22], CPT-v [16], Logit-vd [15], and our RAPMF. We try to answer the following questions:

1. How to collect data with benchmark response patterns to evaluate the models fairly?