

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2015 年 3 月 19 日 (19.03.2015)



(10) 国际公布号
WO 2015/035556 A1

- (51) 国际专利分类号:
G06F 17/30 (2006.01)
- (21) 国际申请号: PCT/CN2013/083218
- (22) 国际申请日: 2013 年 9 月 10 日 (10.09.2013)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (71) 申请人: 华为技术有限公司 (HUAWEI TECHNOLOGIES CO., LTD.) [CN/CN]; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。
- (72) 发明人: 张洪波 (ZHANG, Hongbo); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。 格卢霍夫瓦列里 (Glukhov, Valery); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。
- (74) 代理人: 北京中博世达专利商标代理有限公司 (BEIJING ZBSD PATENT & TRADEMARK AGENT LTD.); 中国北京市海淀区大柳树路 17 号富海大厦 B 座 501 室, Beijing 100081 (CN)。
- (81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。
- (84) 指定国 (除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。
- 本国际公布:
— 包括国际检索报告(条约第 21 条(3))。

(54) Title: RECOMMENDATION METHOD AND DEVICE

(54) 发明名称: 一种推荐方法和设备

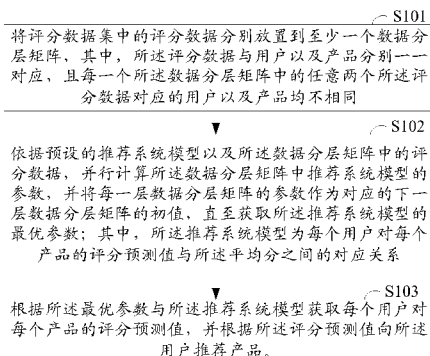


图 1 / Fig. 1

(57) Abstract: Embodiments of the present invention provide a recommendation method and a device. Recommendation efficiency of a recommendation system in a massive data environment is improved through parallel calculation, and a recommendation effect of the recommendation system is improved through considering implicit feedbacks of users, wherein the recommendation method comprises: placing scoring data in a scoring data set to at least one data layer respectively; based on a preset recommendation system model and the scoring data in the data layers, computing parameters of the recommendation system model in the data layers in parallel and using parameters of each layer of the data layers as initial values of a next layer of the data layers until optimal parameters of the recommendation system model are obtained; based on the optimal parameters and the recommendation system model, obtaining a predicted scoring value of each user on each product; and recommending products to the users based on the predicted scoring value.

(57) 摘要:

[见续页]

WO 2015/035556 A1

本发明实施例提供了一种推荐方法和设备，通过并行计算提高了海量数据环境下推荐系统的推荐效率，并且通过考虑用户隐式反馈提高了推荐系统的推荐效果，其中推荐方法包括：将评分数据集中的评分数据分别放置到至少一个数据层；依据预设的推荐系统模型以及数据层中的评分数据，并行计算数据层中推荐系统模型参数，并将每一层数据层的参数作为对应的下一层数据层的初值，直至获取推荐系统模型的最优参数；根据最优参数与推荐系统模型获取每个用户对每个产品的评分预测值，并根据评分预测值向用户推荐产品。

一种推荐方法和设备

技术领域

本发明涉及信息处理领域，尤其涉及一种推荐方法和设备。

背景技术

随着网络的不断发展，网络信息量也呈现了爆炸式的增长，信息推荐方法的出现使得用户能够在海量网络信息中获取自己感兴趣的信息，现有的信息推荐方法是将用户的显示反馈信息与隐式反馈信息融合在数据模型中，通过随机梯度下降算法求解损失函数最小化的优化问题，以串行计算的方式解出数据模型中的参数，并依据此参数对用户尚未评分的产品进行预测并对用户进行推荐预测的用户感兴趣的产品，由于同时考虑了用户的显示反馈和隐式反馈，因此这种方法有较高的推荐精度。

发明人发现现有的信息推荐方法由于采用的是串行计算的方式，并不适合在海量数据环境下进行数据处理，在海量数据环境下效率极低，同时也会影响向用户推荐的效果。

发明内容

本发明实施例提供了一种推荐方法和设备，通过并行计算提高了海量数据环境下推荐系统的推荐效率，并且通过考虑用户隐式反馈提高了推荐系统的推荐效果。

为达到上述目的，本发明的实施例采用如下技术方案：

第一方面，本发明实施例提供了一种推荐方法，包括：

将评分数据集中的评分数据分别放置到至少两个数据层，其中，所述评分数据与用户以及产品分别一一对应，且每一个所述数据层中的任意两个所述评分数据对应的用户以及产品均不相同；

依据预设的推荐系统模型以及所述数据层中的评分数据，并行计算所述数据层中推荐系统模型的参数，并将每一层数据层的参数作为对应的下一层数据层的初值，直至获取所述推荐系统模型的最优参数；其中，所述推荐系统模型为每个用户对每个产品的评分预测值与所述平均分和所述推荐系统模型的参数之间的对应关系；

根据所述最优参数与所述推荐系统模型获取每个用户对每个产品的评分预测值，并根据所述评分预测值向所述用户推荐产品。

根据第一种可能的实现方式，结合第一方面，所述推荐系统模型包括提供了隐式反馈的推荐系统模型、未提供隐式反馈的推荐系统模型、考虑时空特性的推荐系统模型和非对称的潜在因素的推荐系统模型。

根据第二种可能的实现方式，结合第一方面或者第一种可能的实现方式，所述推荐系统包括：

第一推荐系统模型

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j)$$

或者，第二推荐系统模型

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

在所述第一推荐模型和所述第二推荐模型中， $\hat{r}_{ui}$ 表示用户 u 对产品 i 的评分预测值， μ 表示所述评分数据集中的所有评分数据的平均值， b_u 表示所述用户 u 相对用户平均评分的偏移量， b_i 表示所述产品 i 相对产品平均评分的偏移量， q_i 表示产品因素矢量， T 表示转置运算符号， p_u 表示用户因素矢量，

进一步的，在所述第一推荐模型中， $|N(u)|$ 表示用户 u 提供了隐式偏好的所有产品的集合大小， $N(u)$ 表示用户 u 提供了隐式偏好的所有产品的集合； $y_j \in \mathbb{R}$ 表示与产品 j 相关联的因素矢量，其用于表征隐式反馈信息。

根据第三种可能的实现方式，结合第二种可能的实现方式，所述的方法，还包括：

根据所述评分预测值与所述评分数据的均方误差和所述推荐系

统模型的参数之间的关系得到所述推荐系统模型的代价函数，其中所述代价函数包括：

第一代价函数

$$\sum_{(u,i) \in \kappa} [r_{ui} - \mu - b_u - b_i - q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j)]^2 + \lambda_1 (b_u^2 + b_i^2) + \lambda_2 (\|q_i\|^2 + \|p_u\|^2 + \sum_{j \in N(u)} \|y_j\|^2)$$

或者，第二代价函数

$$\sum_{(u,i) \in \kappa} [r_{ui} - \mu - b_u - b_i - q_i^T p_u]^2 + \lambda_1 (b_u^2 + b_i^2) + \lambda_2 (\|q_i\|^2 + \|p_u\|^2)$$

其中， $\|\cdot\|^2$ 表示矢量*的所有元素的平方和， λ_1 与 λ_2 为正则化因子。

根据第四种可能的实现方式，结合第一方面、第一种至第三种可能的实现方式中的任一项，所述依据预设的推荐系统模型以及所述数据层中的评分数据，并行计算所述数据层中推荐系统模型的参数，并将每一层数据层的参数作为对应的下一层数据层的初值，直至获取所述推荐系统模型的最优参数，包括：

A：计算所述评分数据集中的所有所述评分数据的平均分；

B：依次采用并行计算的方式计算每一层所述数据层的参数，并将每一层所述数据层计算所得的参数作为下一层数据层的参数初值；其中，所述第一层所述数据层的参数初值由系统设置；

C：根据最后一层所述数据层计算所得的参数判断所述推荐系统模型是否收敛，若收敛，则计算结束，得到所述最优参数；若不收敛，则将最后一层所述数据层计算所得的参数作为第一层所述数据层的参数初值，重复所述步骤 B、C。

根据第五种可能的实现方式，结合第四种可能的实现方式，所述步骤 B，包括：

B1：根据每一层所述数据层的参数初值和所述推荐系统模型得到所述该层数据层的评分数据的初始估计值，进而根据所述该层数据层的评分数据和所述初始估计值得到所述该层数据层的评分误差；

B2: 根据所述评分误差获取所述该层数据层计算所得的参数;

B3: 将所述该层数据层计算所得的参数作为下一层数据层的参数初值, 根据步骤 B1 和 B2 得到下一层数据层计算所得的参数, 直至得到最后一层所述数据层计算所得的参数。

根据第六种可能的实现方式, 结合第四种或第五种可能的实现方式, 所述根据最后一层所述数据层计算所得的参数判断所述推荐系统模型是否收敛, 包括:

将本次计算所得到的最后一层所述数据层计算所得的参数和前一次计算所得到的最后一层所述数据层计算所得的参数均代入所述代价函数进行计算, 若代入所述代价函数进行计算的结果之差不大于预设的门限值, 则本次计算所得到的最后一层所述数据层计算所得的参数是收敛的, 否则, 本次计算所得到的最后一层所述数据层计算所得的参数是不收敛的。

根据第七种可能的实现方式, 结合第五种可能的实现方式, 所述根据所述评分误差获取所述该层数据层计算所得的参数, 进一步包括:

将所述第一推荐模型中的表达式 $p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ 作为一个等效参数, 并用辅助变量 z_u 来表示所述等效参数, 即 $z_u = p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$;

然后根据所述辅助变量 z_u 的梯度 $\Delta z_{ui} = 2e_{ui} \cdot q_i - \lambda_2 z_u$, 获取所述辅助变量 z_u , 即得到所述等效参数; 以及根据所述辅助变量 z_u 获取所述参数 q_i , 即 $q_i \leftarrow q_i + \gamma_2 (e_{ui} \cdot z_u - \lambda_2 q_i)$, 其中, $\leftarrow$ 符号表示更新符号, 即用更新符号右边的计算值替代更新符号左边的变量值, 更新符号右边出现的参数均为相应的参数的初值, 更新符号左边出现的参数均为参数的更新值。

第二方面, 提供了一种推荐设备, 包括:

数据放置单元, 用于将评分数据集中的评分数据分别放置到至少两个数据层, 其中, 所述评分数据与用户以及产品分别一一对应,

且每一个所述数据层中的任意两个所述评分数据对应的用户以及产品均不相同；

并行计算单元，用于依据预设的推荐系统模型以及所述数据层中的评分数据，并行计算所述数据层中推荐系统模型的参数，并将每一层数据层的参数作为对应的下一层数据层的初值，直至获取所述推荐系统模型的最优参数；其中，所述推荐系统模型为每个用户对每个产品的评分预测值与所述平均分和所述推荐系统模型的参数之间的对应关系；

预测推荐单元，用于根据所述最优参数与所述推荐系统模型获取每个用户对每个产品的评分预测值，并根据所述评分预测值向所述用户推荐产品。

根据第一种可能的实现方式，结合第二方面，所述推荐系统模型包括提供了隐式反馈的推荐系统模型、未提供隐式反馈的推荐系统模型、考虑时空特性的推荐系统模型和非对称的潜在因素的推荐系统模型。

根据第二种可能的实现方式，结合第二方面或者第一种可能的实现方式，所述推荐系统包括：

第一推荐系统模型

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j)$$

或者，第二推荐系统模型

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

在所述第一推荐模型和所述第二推荐模型中， $\hat{r}_{ui}$ 表示用户 u 对产品 i 的评分预测值， μ 表示所述评分数据集中的所有评分数据的平均值， b_u 表示所述用户 u 相对用户平均评分的偏移量， b_i 表示所述产品 i 相对产品平均评分的偏移量， q_i 表示产品因素矢量， T 表示转置运算符号， p_u 表示用户因素矢量，

进一步的，在所述第一推荐模型中， $|N(u)|$ 表示用户 u 提供了隐式偏好的所有产品的集合大小， $N(u)$ 表示用户 u 提供了隐式偏好的所有产品的集合； $y_j \in \mathcal{F}$ 表示与产品 j 相关联的因素矢量，其用于表征隐式反馈信息。

根据第三种可能的实现方式，结合第二种可能的实现方式，还包括：代价函数生成单元，用于根据所述评分预测值与所述评分数据的均方误差和所述推荐系统模型的参数之间的关系得到所述推荐系统模型的代价函数，其中所述代价函数包括：

第一代价函数

$$\sum_{(u,i) \in \kappa} [r_{ui} - \mu - b_u - b_i - q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j)]^2 + \lambda_1 (b_u^2 + b_i^2) + \lambda_2 (\|q_i\|^2 + \|p_u\|^2 + \sum_{j \in N(u)} \|y_j\|^2)$$

或者，第二代价函数

$$\sum_{(u,i) \in \kappa} [r_{ui} - \mu - b_u - b_i - q_i^T p_u]^2 + \lambda_1 (b_u^2 + b_i^2) + \lambda_2 (\|q_i\|^2 + \|p_u\|^2)$$

其中， $\|\cdot\|^2$ 表示矢量*的所有元素的平方和， λ_1 与 λ_2 为正则化因子。

根据第四种可能的实现方式，结合第二方面、第一种至第三种可能的实现方式中的任一项，所述并行计算单元，包括：

平均分计算子单元，用于计算所述评分数据集中的所有所述评分数据的平均分；

分层计算子单元，用于依次采用并行计算的方式计算每一层所述数据层的参数，并将每一层所述数据层计算所得的参数作为下一层数据层的参数初值；其中，所述第一层所述数据层的参数初值由系统设置；

收敛判断子单元，用于根据最后一层所述数据层计算所得的参数判断所述推荐系统模型是否收敛，若收敛，则计算结束，得到所述最优参数；若不收敛，则将最后一层所述数据层计算所得的参数作为第一层所述数据层的参数初值，并将所述参数初值传输至所述分层计算子单元重复进行分层计算。

根据第五种可能的实现方式，结合第四种可能的实现方式，所述分层计算子单元进一步用于，

评分误差生成模块，用于根据每一层所述数据层的参数初值和所述推荐系统模型得到所述该层数据层的评分数据的初始估计值，

进而根据所述该层数据层的评分数据和所述初始估计值得到所述该层数据层的评分误差；

参数计算模块，用于根据所述评分误差获取所述该层数据层计算所得的参数；

计算控制模块，用于将所述该层数据层计算所得的参数作为下一层数据层的参数初值，通过所述评分误差生成模块和所述参数计算模块得到下一层数据层计算所得的参数，直至得到最后一层所述数据层计算所得的参数。

根据第六种可能的实现方式，结合第四种或第五种可能的实现方式，所述收敛判断子单元进一步用于，将本次计算所得到的最后一层所述数据层计算所得的参数和前一次计算所得到的最后一层所述数据层计算所得的参数均代入所述代价函数进行计算，若代入所述代价函数进行计算的结果之差不大于预设的门限值，则本次计算所得到的最后一层所述数据层计算所得的参数是收敛的，否则，本次计算所得到的最后一层所述数据层计算所得的参数是不收敛的。

根据第七种可能的实现方式，结合第五种可能的实现方式，所述参数计算模块进一步用于，

将所述第一推荐模型中的表达式 $p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ 作为一个等效参数，并用辅助变量 z_u 来表示所述等效参数，即 $z_u = p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ ；

然后根据所述辅助变量 z_u 的梯度 $\Delta z_{ui} = 2e_{ui} \cdot q_i - \lambda_2 z_u$ ，获取所述辅助变量 z_u ，即得到所述等效参数；以及根据所述辅助变量 z_u 获取所述参数 q_i ，即 $q_i \leftarrow q_i + \gamma_2 (e_{ui} \cdot z_u - \lambda_2 q_i)$ ，其中， $\leftarrow$ 符号表示更新符号，即用更新符号右边的计算值替代更新符号左边的变量值，更新符号右边出现的参数均为相应的参数的初值，更新符号左边出现的参数均为参数的更新值。

第三方面，提供了一种推荐设备，包括处理器和存储器，其中，

所述处理器用于，将评分数据集中的评分数据分别放置到至少两个数据层，其中，所述评分数据与用户以及产品分别一一对应，

且每一个所述数据层中的任意两个所述评分数据对应的用户以及产品均不相同；

以及依据预设的推荐系统模型以及所述数据层中的评分数据，并行计算所述数据层中推荐系统模型的参数，并将每一层数据层的参数作为对应的下一层数据层的初值，直至获取所述推荐系统模型的最优参数；其中，所述推荐系统模型为每个用户对每个产品的评分预测值与所述平均分和所述推荐系统模型的参数之间的对应关系；

以及根据所述最优参数与所述推荐系统模型获取每个用户对每个产品的评分预测值，并根据所述评分预测值向所述用户推荐产品；

所述存储器用于保存评分数据集以及处理器所执行的程序和执行的结果。

根据第一种可能的实现方式，结合第三方面，所述推荐系统模型包括提供了隐式反馈的推荐系统模型、未提供隐式反馈的推荐系统模型、考虑时空特性的推荐系统模型和非对称的潜在因素的推荐系统模型。

根据第二种可能的实现方式，结合第三方面或者第一种可能的实现方式，所述推荐系统包括：

第一推荐系统模型

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j)$$

或者，第二推荐系统模型

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

在所述第一推荐模型和所述第二推荐模型中， $\hat{r}_{ui}$ 表示用户 u 对产品 i 的评分预测值， μ 表示所述评分数据集中的所有评分数据的平均值， b_u 表示所述用户 u 相对用户平均评分的偏移量， b_i 表示所述产品 i 相对产品平均评分的偏移量， q_i 表示产品因素矢量， T 表示转置运算符号， p_u 表示用户因素矢量，

进一步的，在所述第一推荐模型中， $|N(u)|$ 表示用户 u 提供了隐式偏好的所有产品的集合大小， $N(u)$ 表示用户 u 提供了隐式偏好的所

有产品的集合； $y_j \in \mathcal{F}$ 表示与产品j相关联的因素矢量，其用于表征隐式反馈信息。

根据第三种可能的实现方式，结合第二种可能的实现方式，所述处理器还用于，根据所述评分预测值与所述评分数据的均方误差和所述推荐系统模型的参数之间的关系得到所述推荐系统模型的代价函数，其中所述代价函数包括：

第一代价函数

$$\sum_{(u,i) \in \kappa} [r_{ui} - \mu - b_u - b_i - q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j)]^2 + \lambda_1 (b_u^2 + b_i^2) + \lambda_2 (\|q_i\|^2 + \|p_u\|^2 + \sum_{j \in N(u)} \|y_j\|^2)$$

或者，第二代价函数

$$\sum_{(u,i) \in \kappa} [r_{ui} - \mu - b_u - b_i - q_i^T p_u]^2 + \lambda_1 (b_u^2 + b_i^2) + \lambda_2 (\|q_i\|^2 + \|p_u\|^2)$$

其中， $\|\cdot\|^2$ 表示矢量*的所有元素的平方和， λ_1 与 λ_2 为正则化因子。

根据第四种可能的实现方式，结合第三方面、第一种至第三种可能的实现方式中的任一项，所述处理器用于，

A：计算所述评分数据集中的所有所述评分数据的平均分；

B：依次采用并行计算的方式计算每一层所述数据层的参数，并将每一层所述数据层计算所得的参数作为下一层数据层的参数初值；其中，所述第一层所述数据层的参数初值由系统设置；

C：根据最后一层所述数据层计算所得的参数判断所述推荐系统模型是否收敛，若收敛，则计算结束，得到所述最优参数；若不收敛，则将最后一层所述数据层计算所得的参数作为第一层所述数据层的参数初值，重复所述步骤B、C。

根据第五种可能的实现方式，结合第四种可能的实现方式，所述处理器进一步用于，

B1：根据每一层所述数据层的参数初值和所述推荐系统模型得到所述该层数据层的评分数据的初始估计值，进而根据所述该层数据层的评分数据和所述初始估计值得到所述该层数据层的评分误差；

B2: 根据所述评分误差获取所述该层数据层计算所得的参数;

B3: 将所述该层数据层计算所得的参数作为下一层数据层的参数初值, 根据步骤 B1 和 B2 得到下一层数据层计算所得的参数, 直至得到最后一层所述数据层计算所得的参数。

根据第六种可能的实现方式, 结合第四种或第五种可能的实现方式, 所述处理器用于,

将本次计算所得到的最后一层所述数据层计算所得的参数和前一次计算所得到的最后一层所述数据层计算所得的参数均代入所述代价函数进行计算, 若代入所述代价函数进行计算的结果之差不大于预设的门限值, 则本次计算所得到的最后一层所述数据层计算所得的参数是收敛的, 否则, 本次计算所得到的最后一层所述数据层计算所得的参数是不收敛的。

根据第七种可能的实现方式, 结合第五种可能的实现方式, 所述处理器用于根据所述评分误差获取所述该层数据层计算所得的参数, 进一步包括:

所述处理器将所述第一推荐模型中的表达式 $p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ 作为一个等效参数, 并用辅助变量 z_u 来表示所述等效参数, 即 $z_u = p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$;

然后根据所述辅助变量 z_u 的梯度 $\Delta z_{ui} = 2e_{ui} \cdot q_i - \lambda_2 z_u$, 获取所述辅助变量 z_u , 即得到所述等效参数; 以及根据所述辅助变量 z_u 获取所述参数 q_i , 即 $q_i \leftarrow q_i + \gamma_2 (e_{ui} \cdot z_u - \lambda_2 q_i)$, 其中, $\leftarrow$ 符号表示更新符号, 即用更新符号右边的计算值替代更新符号左边的变量值, 更新符号右边出现的参数均为相应的参数的初值, 更新符号左边出现的参数均为参数的更新值。

本发明实施例提供的推荐方法和设备, 通过并行计算提高了海量数据环境下推荐系统的推荐效率, 并且通过考虑用户隐式反馈提高了推荐系统的推荐效果。

附图说明

为了更清楚地说明本发明实施例或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本发明的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

图 1 为本发明实施例提供一种推荐方法流程示意图；

图 2 为本发明实施例提供一种推荐方法的详细流程图；

图 3A 为本发明实施例所提供的一种评分数据的放置方法流程图；

图 3B 为本发明实施例所提供的另一种评分数据的放置方法流程图；

图 4 为本发明实施例所提供的最优参数的求解流程示意图；

图 5A 为本发明实施例所提供的一种参数更新方法的示意图；

图 5B 为本发明实施例所提供的另一种参数更新方法的示意图；

图 6 为本发明实施例提供一种推荐设备结构图；

图 7 为本发明实施例提供的另一种推荐设备的结构图；

图 8 为本发明实施例提供一种推荐设备的硬件装置图。

具体实施方式

下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本发明一部分实施例，而不是全部的实施例。基于本发明中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本发明保护的范围。

如图 1 所示，为本发明实施例提供一种推荐方法的流程示意图，包括：

S101: 将评分数据集中的评分数据分别放置到至少两个数据层，

其中，评分数据与用户以及产品分别一一对应，且每一个数据层中的任意两个评分数据对应的用户以及产品均不相同；

S102: 依据预设的推荐系统模型以及数据层中的评分数据，并行计算数据层中推荐系统模型的参数，并将每一层数据层的参数作为对应的下一层数据层的初值，直至获取推荐系统模型的最优参数；其中，推荐系统模型为每个用户对每个产品的评分预测值与平均分和推荐系统模型的参数之间的对应关系；

示例性的，推荐系统模型可以包括提供了隐式反馈的推荐系统模型、未提供隐式反馈的推荐系统模型、考虑时空特性的推荐系统模型和非对称的潜在因素的推荐系统模型；

示例性的，推荐系统模型还可以包括：

第一推荐系统模型

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j)$$

或者，第二推荐系统模型

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

在第一推荐模型和第二推荐模型中， $\hat{r}_{ui}$ 表示用户 u 对产品 i 的评分预测值， μ 表示评分数据集中的所有评分数据的平均值， b_u 表示用户 u 相对用户平均评分的偏移量， b_i 表示产品 i 相对产品平均评分的偏移量， q_i 表示产品因素矢量， T 表示转置运算符号， p_u 表示用户因素矢量；

进一步的，在第一推荐模型中， $|N(u)|$ 表示用户 u 提供了隐式偏好的所有产品的集合大小， $N(u)$ 表示用户 u 提供了隐式偏好的所有产品的集合； $y_j \in \mathcal{F}$ 表示与产品 j 相关联的因素矢量，其用于表征隐式反馈信息。

相应的，根据推荐系统模型得到的预测值与评分数据的均方误差和推荐系统模型的参数之间的关系得到推荐系统模型的代价函数，其中代价函数包括：

第一代价函数

$$\sum_{(u,i) \in \kappa} [r_{ui} - \mu - b_u - b_i - q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j)]^2 + \lambda_1 (b_u^2 + b_i^2) + \lambda_2 (\|q_i\|^2 + \|p_u\|^2 + \sum_{j \in N(u)} \|y_j\|^2)$$

或者，第二代价函数

$$\sum_{(u,i) \in \kappa} [r_{ui} - \mu - b_u - b_i - q_i^T p_u]^2 + \lambda_1 (b_u^2 + b_i^2) + \lambda_2 (\|q_i\|^2 + \|p_u\|^2)$$

其中， $\|\cdot\|^2$ 表示矢量*的所有元素的平方和， λ_1 与 λ_2 为正则化因子。

示例性的，依据预设的推荐系统模型以及数据层中的评分数据，并行计算数据层中推荐系统模型的参数，并将每一层数据层的参数作为对应的下一层数据层的初值，直至获取推荐系统模型的最优参数，包括：

A：计算评分数据集中的所有评分数据的平均分；

B：依次采用并行计算的方式计算每一层数据层的参数，并将每一层数据层计算所得的参数作为下一层数据层的参数初值；其中，第一层数据层的参数初值由系统设置；

进一步的，步骤 B，包括：

B1：根据每一层数据层的参数初值和推荐系统模型得到该层数据层的评分数据的初始估计值，进而根据该层数据层的评分数据和初始估计值得到该层数据层的评分误差；

B2：根据评分误差获取该层数据层计算所得的参数；

B3：将该层数据层计算所得的参数作为下一层数据层的参数初值，根据步骤 B1 和 B2 得到下一层数据层计算所得的参数，直至得到最后一层数据层计算所得的参数。

C：根据最后一层数据层计算所得的参数判断推荐系统模型是否收敛，若收敛，则计算结束，得到最优参数；得到最优参数，若不收敛，则将最后一层数据层计算所得的参数作为第一层数据层的参数初值，重复步骤 B、C。

示例性的，根据最后一层数据层计算所得的参数判断推荐系统模型是否收敛，包括：

将本次计算所得到的最后一层数据层计算所得的参数和前一次计算所得到的最后一层数据层计算所得的参数均代入代价函数进行计算，若代入代价函数进行计算的结果之差不大于预设的门限值，则本次计算所得到的最后一层数据层计算所得的参数是收敛的，否则，本次计算所得到的最后一层数据层计算所得的参数是不收敛的。

S103: 根据最优参数与推荐系统模型获取每个用户对每个产品的评分预测值，并根据评分预测值向用户推荐产品；

本实施例提供了一种推荐方法，通过并行计算提高了海量数据环境下推荐系统的推荐效率，并且通过考虑用户隐式反馈提高了推荐系统的推荐效果。

如图 2 所示，为本实施例提供的一种推荐方法的详细流程图，包括；

S201: 将评分数据集中的评分数据分别放置到至少两个数据层；

示例性的，评分数据集可以通过获取用户对产品的评分数据得到，也可以通过用户的浏览、购买记录信息得到，不仅可以得到用户对产品的显示反馈，也可以获取到用户偏好的隐式反馈，本发明实施例对此不做任何限制，优选的，评分数据集可以通过用户对产品的评分数据获得，本领域技术人员可以理解的，用户对产品的评分数据不仅显示的反馈了用户对产品的评价，而且也通过用户对产品评分的行为隐式的反馈了用户对产品的偏好，优选的，在本实施例中，评分数据集可以通过矩阵形式表示，其中，矩阵的不同行表示不同用户，矩阵的不同列表示不同产品；

进一步的，评分数据与用户以及产品分别一一对应，本领域技术人员可以理解的，由于评分数据所放置的数据层可以通过矩阵的形式进行表示，同时评分数据集也可以通过矩阵的形式进行表示，其中，矩阵的不同行表示不同用户，矩阵的不同列表示不同产品，于是可以将评分数据集中的评分数据分别放置到至少两个数据层矩阵，并且所有数据层矩阵均与评分数据集的矩阵具有相同的行数和相同的列数，当每一个数据层中的任意两个评分数据对应的用户以

及产品均不相同，即每一个数据层矩阵中的任意两个评分数据均不在同一行且均不在同一列的时候，就可以满足同一层数据层中的所有评分数据之间互相没有依赖关系，因此可以对同一层数据层中的评分数据进行并行计算，具体的放置步骤本发明实施例不作任何限定，任何能够使得每一个数据层矩阵中的任意两个评分数据均不在同一行且均不在同一列的放置方法均在本发明实施例的保护范围内；

可选的，如图 3A 所示，本实施例优选的一种放置方法包括：

3A01：选取评分数据集中的—个评分数据，将其放在第—层数据层矩阵中对应于该评分数据的用户和产品的位置上，此时，最后—层数设置为 $l_{\max}=1$ ，其中，选取的方式本实施例不做限定，后续从评分数据集中选取数据的方式与第—次选取的方式相同，在此不作赘述；

3A02：选取下一个评分数据，并从第—层数据层矩阵开始到最后一层对应的数据层矩阵，依次与其中的所有评分数据进行比较，是否满足评分数据在数据层矩阵中的对应位置所在的行和列均没有评分数据，若满足，则该评分数据放置于最先满足的数据层矩阵，若直至最后一层 $l_{\max}$ 对应的数据层矩阵也不满足，则最后一层数更新为 $l_{\max} \leftarrow l_{\max} + 1$ ，其中 $\leftarrow$ 符号表示将符号左边的数据更新为符号右边的数据，下同，并且将该评分数据放置于更新后的最后一层数据层；

3A03：对评分数据集中剩余的评分数据依次重复步骤 302 直至评分数据集中所有的评分数据均放置完毕。

例如，评分数据集为：

$$A = \begin{bmatrix} 3 & * & * & 2 & * & * \\ * & 2 & * & 2 & 1 & * \\ 4 & * & 5 & * & 5 & 2 \\ * & * & 2 & * & 3 & * \\ * & 1 & * & * & * & 5 \\ 1 & 2 & * & 3 & * & * \\ * & * & 3 & 5 & 2 & * \end{bmatrix},$$

通过如图 3A 所示的放置方法，示例性的，具体操作过程如下：评分数据集中选出一个评分数据 r_{ui} ，将其放入第一层，然后再选择第二个评分数据 $r_{u'i'}$ ，判断 u' 和 u 是否为同一个用户以及 i' 和 i 是否为同一个产品，当 $u' \neq u$ 且 $i' \neq i$ 时，该评分数据 $r_{u'i'}$ 放入第一层，否则将其放入第二层。随后再取出第三个评分数据 $r_{u''i''}$ ，如果该评分数据所对应的用户 u'' 和产品 i'' 与放置在第一层的用户和产品均不相同时，则将其放入第一层，否则与第二层的评分数据对应的用户和产品进行比较，当满足 $u'' \neq u_{layer2}$ 且 $i'' \neq i_{layer2}$ 时，则将其放入第二层，否则放入第三层，这里 u_{layer2} 表示第二层的用户， i_{layer2} 表示第二层的产品。依此类推，直到集合 κ 的所有评分数据均放置完毕。

可以依次得到 5 个数据层，从第一层数据层到最后一层也就是第五层数据层依次为：

$$\begin{aligned}
 &(u_1, i_1, 3) \ (u_2, i_2, 2) \ (u_3, i_3, 5) \ (u_4, i_5, 3) \ (u_5, i_6, 5) \ (u_6, i_4, 3) \quad l_1(\text{Layer } 1) \\
 &(u_1, i_4, 2) \ (u_2, i_5, 1) \ (u_3, i_1, 4) \ (u_4, i_3, 2) \ (u_5, i_2, 1) \quad l_2(\text{Layer } 2) \\
 &(u_2, i_4, 2) \ (u_3, i_5, 5) \ (u_6, i_1, 1) \ (u_7, i_3, 3) \quad l_3(\text{Layer } 3) \\
 &(u_3, i_6, 2) \ (u_6, i_2, 2) \ (u_7, i_4, 5) \quad l_4(\text{Layer } 4) \\
 &(u_7, i_5, 2) \quad l_5(\text{Layer } 5)
 \end{aligned}$$

通过矩阵形式表示可以是：

$$\begin{aligned}
 l_1 &= \begin{bmatrix} 3 & * & * & * & * & * \\ * & 2 & * & * & * & * \\ * & * & 5 & * & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 5 \\ * & * & * & 3 & * & * \\ * & * & * & * & * & * \end{bmatrix}, \quad l_2 = \begin{bmatrix} * & * & * & 2 & * & * \\ * & * & * & * & 1 & * \\ 4 & * & * & * & * & * \\ * & * & 2 & * & * & * \\ * & 1 & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}, \quad l_3 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & 2 & * & * \\ * & * & * & * & 5 & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ 1 & * & * & * & * & * \\ * & * & 3 & * & * & * \end{bmatrix}, \\
 l_4 &= \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & 2 \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & 2 & * & * & * & * \\ * & * & * & 5 & * & * \end{bmatrix}, \quad l_5 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & 2 & * \end{bmatrix}
 \end{aligned}$$

其中, 矩阵中出现的*表示该位置对应的用户没有对该位置对应的产品进行评分, 可以得到 $A=l_1+l_2+l_3+l_4+l_5$ 。

可选的, 如图 3B 所示, 本实施例优选的另一种放置方法包括:

3B01: 选取评分数据集中的第一个评分数据, 将其放在第一层数据层中对应于该评分数据的用户和产品的位置上, 此时, 最后一层数 $l_{\max}=1$,

3B02: 从评分数据集中选取的第二个评分数据开始, 将该评分数据依次从最后一层数据层直至第一层数据层进行比较, 直至能够从最后一层到第一层数据层中找出最后一个满足该评分数据在数据层矩阵中对应位置的所在行和所在列上没有其他评分数据, 则该评分数据放置于最后一个满足的数据层的对应位置, 若从最后一层数到第一层均没有满足的数据评分矩阵, 则最后一层数更新为 $l_{\max} \leftarrow l_{\max} + 1$, 并且将该评分数据放置于更新后的最后一层数对应的数据层;

3B03: 对评分数据集中剩余的评分数据依次重复步骤 3B02 直至评分数据集中所有的评分数据均放置完毕。

例如, 仍以矩阵 A 为例, 通过如图 3B 所示的放置方法, 示例性的, 具体操作过程如下:

从评分数据集中选出一个评分数据 r_{ui} , 将其放入第一层, 然后再选择第二个评分数据 $r_{u'i'}$, 判断 u' 和 u 是否为同一个用户以及 i' 和 i 是否为同一个产品, 当 $u' \neq u$ 且 $i' \neq i$ 时, 该评分数据 $r_{u'i'}$ 放入第一层, 否则将其放入第二层。随后再取出第三个评分数据 $r_{u''i''}$, 将其与第二层已放置的评分数据进行比较 (如果有第二层的话), 如果该评分数据所对应的用户 u'' 和产品 i'' 与放置在第二层的用户 u' 和产品 i' 均不相同时, 则继续将其与第一层已放置的评分数据进行比较, 当与第一层已放置的评分数据也满足 $u'' \neq u$ 且 $i'' \neq i$ 时则将其放入第一层, 否则放置第二层, 而当该评分数据所对应的用户 u'' 和产品 i'' 与放置在第二层的用户 u' 和产品 i' 满足 $u'' = u'$ 或者 $i'' = i'$ 任意一个条件时, 则直接将其放入第三层。依此类推, 直到评分数据集中的所有评分数据均放置完毕, 可以依次得到 7 个数据层, 从第一层数据层到最后一层也就是第七层数据层依次为:

$(u_1, i_1, 3)$	$(u_2, i_2, 2)$		$l_1(\text{Layer } 1)$
$(u_1, i_4, 2)$	$(u_3, i_1, 4)$	$(u_5, i_2, 1)$	$l_2(\text{Layer } 2)$
$(u_2, i_4, 2)$	$(u_3, i_3, 5)$	$(u_6, i_1, 1)$	$l_3(\text{Layer } 3)$
$(u_2, i_5, 1)$	$(u_4, i_3, 2)$	$(u_6, i_2, 2)$	$l_4(\text{Layer } 4)$
$(u_3, i_5, 5)$	$(u_6, i_4, 3)$	$(u_7, i_3, 3)$	$l_5(\text{Layer } 5)$
$(u_3, i_6, 2)$	$(u_4, i_5, 3)$	$(u_7, i_4, 5)$	$l_6(\text{Layer } 6)$
$(u_5, i_6, 5)$	$(u_7, i_5, 2)$		$l_7(\text{Layer } 7)$

通过矩阵形式表示为：

$$\begin{aligned}
 l_1 &= \begin{bmatrix} 3 & * & * & * & * & * \\ * & 2 & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}, \quad l_2 = \begin{bmatrix} * & * & * & 2 & * & * \\ * & * & * & * & * & * \\ 4 & * & * & * & * & * \\ * & * & * & * & * & * \\ * & 1 & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}, \quad l_3 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & 2 & * & * \\ * & * & 5 & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ 1 & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}, \\
 l_4 &= \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & 1 & * \\ * & * & * & * & * & * \\ * & * & 2 & * & * & * \\ * & * & * & * & * & * \\ * & 2 & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}, \quad l_5 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & 5 & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & 3 & * & * & * \end{bmatrix}, \quad l_6 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & 2 \\ * & * & * & * & 3 & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & 5 & * & * \end{bmatrix}, \\
 l_7 &= \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & 5 \\ * & * & * & * & * & * \\ * & * & * & * & 2 & * \end{bmatrix}
 \end{aligned}$$

其中，矩阵中出现的*表示该位置对应的用户没有对该位置对应的产品进行评分，可以得到 $A=l_1+l_2+l_3+l_4+l_5+l_6+l_7$ 。

S202：依据预设的推荐系统模型以及数据层中的评分数据，并

行计算数据层中推荐系统模型的参数，并将每一层数据层的参数作为对应的下一层数据层的初值，直至获取推荐系统模型的最优参数；

示例性的，预设的系统模型可以是考虑用户隐式反馈的潜在因素模型，也可以是仅仅基于现实反馈的潜在因素模型，还可以包括考虑时空特性的推荐系统模型、非对称的潜在因素模型等，本发明实施例对此不做任何限定，

进一步的，在本实施例中，构建了一个改进的考虑用户隐式反馈的第一推荐系统模型，如式（1），以及一个未考虑用户隐式反馈的第二推荐系统模型，如式（2）：

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j) \quad (1)$$

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u \quad (2)$$

式（1）和式（2）中， $\hat{r}_{ui}$ 表示用户 u 对产品 i 的评分预测值， μ 表示评分数据集中的所有评分数据的平均值， b_u 表示用户 u 相对用户平均评分的偏移量， b_i 表示产品 i 相对产品平均评分的偏移量， q_i 表示产品因素矢量， T 表示转置运算符号， p_u 表示用户因素矢量；

进一步在式（1）中， $|N(u)|$ 表示用户 u 提供了隐式偏好的所有产品的集合大小， $N(u)$ 表示用户 u 提供了隐式偏好的所有产品的集合； $y_j \in \mathbb{R}^f$ 表示与产品 j 相关联的因素矢量，其用于表征隐式反馈信息。而且 b_u ， b_i ， q_i ， p_u 和 y_j 为用户的推荐系统模型的未知参数；

示例性的，根据推荐系统模型得到的预测值与评分数据的均方误差和推荐系统模型的参数之间的关系得到推荐系统模型的代价函数，以及根据步骤 S201 得到的分层数据矩阵的评分数据，通过对上述模型有关的代价函数最优化问题的求解得到上述模型的未知参数，即并行计算数据层中推荐系统模型的参数，并将每一层数据层的参数作为对应的下一层数据层的初值，直至获取推荐系统模型的最优参数，其中，最优参数就是上述模型的未知参数最优值，具体的，在本实施例中，第一推荐系统模型和第二推荐系统模型有关的代价函数可以分别表示为式（3）的第一代价函数和式（4）的第二代价函数：

$$\sum_{(u,i) \in \kappa} [r_{ui} - \mu - b_u - b_i - q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j)]^2 + \lambda_1 (b_u^2 + b_i^2) + \lambda_2 (\|q_i\|^2 + \|p_u\|^2 + \sum_{j \in N(u)} \|y_j\|^2) \quad (3)$$

$$\sum_{(u,i) \in \kappa} [r_{ui} - \mu - b_u - b_i - q_i^T p_u]^2 + \lambda_1 (b_u^2 + b_i^2) + \lambda_2 (\|q_i\|^2 + \|p_u\|^2) \quad (4)$$

其中, $\|*\|^2$ 表示矢量*的所有元素的平方和, λ_1 与 λ_2 为正则化因子。

本领域技术人员可以理解的, 对式(3)与式(4)进行最优化问题的求解, 具体的求解过程具有相似性, 不再赘述, 如图4所示, 对于式(3)进行最优化问题的求解, 可以包括:

401、计算评分数据集中的所有评分数据的平均分 μ ;

402、依次采用并行计算的方式计算每一层数据层的参数, 并将每一层数据层计算所得的参数作为下一层数据层的参数初值;

示例性的, 本实施例中, 在第一次计算之前, 可以将第一层数据层的参数 b_u , b_i , q_i , p_u 和 y_j 以及 λ_1 与 λ_2 的初值进行随机设置, 为了简便起见, 参数 b_u , b_i , q_i , p_u 和 y_j 的初值可以设置为标量参数为0, 矢量参数为向量 $\vec{0}$, 其中, 数字0上方的箭头符号为向量符号, 而 λ_1 与 λ_2 可以任意设置为一个为比较小的正值, 本发明实施例对此不作任何限定, 用来表示正则化因子;

示例性的, 从第一层数据层开始, 依次的并行计算该层数据层的参数 b_u , b_i , q_i , p_u 和 y_j 的值, 并将计算所得的参数值作为下一层数据层的参数 b_u , b_i , q_i , p_u 和 y_j 的初值, 直至计算完最后一层数据层的参数 b_u , b_i , q_i , p_u 和 y_j , 得到最后一层数据层计算所得的参数; 具体的计算过程如下:

根据第一层数据层的参数初值对第一层数据层所包含的评分数据对应的未知参数 b_u , b_i , q_i , p_u 和 y_j 进行更新计算, 由于同一层矩阵的评分数据之间没有相互依赖的关系, 因此同一层矩阵的不同评分数据对应的未知参数 b_u , b_i , q_i , p_u 和 y_j 的更新计算可以并行进行, 并将更新后的参数作为下一层数据层的参数初值对下一层数据层所包含的评分数据对应的参数进行更新计算, 直至最后一层数据层所包含的评分数据对应的参数进行更新计算完毕, 得到最后一层数据

层计算所得的参数；

优选的，在本实施例中，可以通过梯度下降法进行更新计算，由于所有评分数据对应的参数的更新方法均相同，本领域技术人员可以理解，通过对一个评分数据 r_{ui} 对应的参数 b_u , b_i , q_i , p_u 和 y_j 进行更新计算步骤进行描述后，可以无需创造性的将计算步骤应用在其他评分数据中，如图 4 所示，具体的步骤如下：

4021、根据每一层数据层的参数初值和式 (1) 所示的推荐系统模型得到评分数据 r_{ui} 的初始估计值 $\hat{r}_{ui}$ ，进而根据该层数据层的评分数据和初始估计值 $\hat{r}_{ui}$ 得到该层数据层的评分误差 e_{ui} ；

4022、根据评分误差 e_{ui} 得到负梯度方向的该层数据层计算所得的参数的更新值，如图 5A 所示的示意图，包括：

$$b_u \leftarrow b_u + \gamma_1 \cdot (e_{ui} - \lambda_1 b_u) \quad (5)$$

$$b_i \leftarrow b_i + \gamma_1 \cdot (e_{ui} - \lambda_1 b_i) \quad (6)$$

$$p_u \leftarrow p_u + \gamma_2 \cdot (e_{ui} \cdot q_i - \lambda_2 p_u) \quad (7)$$

$$q_i \leftarrow q_i + \gamma_2 \cdot \left(e_{ui} \cdot \left(p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j \right) - \lambda_2 q_i \right) \quad (8)$$

$$y_j \leftarrow y_j + \gamma_2 \cdot \left(e_{ui} \cdot |N(u)|^{-\frac{1}{2}} \cdot q_i - \lambda_2 y_j \right), \quad \forall j \in N(u) \quad (9)$$

其中， $\leftarrow$ 符号表示更新符号，即用更新符号右边的计算值替代更新符号左边的变量值，在本实施例中，式 (5) - 式 (9) 的更新符号右边出现的参数均为相应的参数的初值，更新符号左边出现的参数均为参数的更新值， γ_1 和 γ_2 为迭代步长，本实施例中优选的，可以是一个设定的合适的常量，即可以设定 γ_1 和 γ_2 为某个常量，或者也可以是一个与迭代次数相关的量，即随着迭代次数的增加而逐渐减少迭代步长。如 $\gamma_1 * 0.96^{\text{NumOfIter}}$ 和 $\gamma_2 * 0.96^{\text{NumOfIter}}$ ，其中，NumOfIter 表示迭代次数。本发明实施例对此不做限定。

本领域技术人员可以理解的，式 (5) - 式 (8) 均可以根据相应的计算式进行参数的更新，计算过程不再赘述，相应的，对式 (4)

的最优化问题的求解只需要进行式 (5) - 式 (8) 的更新计算, 得到参数 b_u , b_i , q_i , p_u 即可;

但是对于式 (9) 的计算还需要结合评分数据集中的评分数据依次对 y_j 进行串行计算, 降低了运算效率, 因此本发明实施例提供了一种针对式 (9) 的计算方法, 如图 5A 所示的示意图, 具体实施方式为:

当同一层矩阵有多个 y_j 需要计算时, 利用评分数据集中 y_j 所表示的产品 j 对应的用户来并行计算其对应的梯度 Δy_j , 然后将产品 j 对应的所有的梯度 Δy_j 进行聚合得到该层 y_j 的值, 比如在评分数据集中, 用户 1, 用户 2, 和用户 6 对产品 4 ($j=4$) 进行了评分, 则可以并行计算用户 1, 用户 2, 和用户 6 对应的产品 j 的梯度 $\Delta y_j^{(1)}$, $\Delta y_j^{(2)}$, $\Delta y_j^{(6)}$, 其中, $\Delta y_j^{(1)} \leftarrow e_{11} \cdot 2^{-\frac{1}{2}} \cdot q_1 - \lambda_2 y_j^0$, $\Delta y_j^{(2)} \leftarrow e_{22} \cdot 3^{-\frac{1}{2}} \cdot q_2 - \lambda_2 y_j^0$, $\Delta y_j^{(6)} \leftarrow e_{64} \cdot 3^{-\frac{1}{2}} \cdot q_4 - \lambda_2 y_j^0$;

然后通过 $y_j \leftarrow y_j^o + \gamma_2 (\Delta y_j^{(1)} + \Delta y_j^{(2)} + \Delta y_j^{(6)})$ 得到该层 y_j 的更新值, 其中, y_j^o 为 y_j 在该层计算时的初值。

此外, 本发明实施例提供了另一种针对式 (8) 和式 (9) 的等效计算方法, 如图 5B 所示的示意图, 具体实施方式为:

将第一推荐模型中的表达式 $p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ 作为一个等效参数, 并用辅助变量 z_u 来表示所述等效参数, 即 $z_u = p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$, 其中辅助变量 z_u 与表达式 $p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ 等价, 此时, 第一推荐模型等价于 $\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot z_u$, 其参数为 b_u , b_i , q_i , z_u ;

然后根据辅助变量 z_u 的梯度 $\Delta z_{ui} = 2e_{ui} \cdot q_i - \lambda_2 z_u$, 并由此得到辅助变量 z_u 的更新公式 $z_u \leftarrow z_u^0 + \gamma_2 \cdot (2e_{ui} \cdot q_i - \lambda_2 z_u^0)$, 其中 z_u^0 为 z_u 在该层计算时的初值。

通过本方法, 可以只通过对辅助变量 z_u 的更新过程来替代对参数 p_u 和 y_j 的更新过程, 此外, 由于引入了辅助变量 z_u , 式 (8) 的关于参数 q_i 的更新也可以做出相应的变动, 即 q_i 的更新公式变为

$q_i \leftarrow q_i^0 + \gamma_2 \cdot (e_{ui} \cdot z_u - \lambda_2 q_i^0)$, 其中 q_i^0 为 q_i 在该层计算时的初值。

由此可知, 引入辅助变量 z_u 一方面简化了计算, 另一方面由于消除了 y_j 的计算, 推荐模型的参数求解减少了一层内循环, 从而在确保同样精度的前提下极大地提升了运算速度。

4023, 将该层数据层计算所得的参数作为下一层数据层的参数初值, 根据步骤 4021 和 4022 得到下一层数据层计算所得的参数, 直至得到最后一层数据层计算所得的参数。

403: 根据最后一层数据层计算所得的参数判断推荐系统模型是否收敛, 若收敛, 则计算结束, 得到最优参数; 得到最优参数, 若不收敛, 则将最后一层数据层计算所得的参数作为第一层数据层的参数初值, 重复步骤 402, 403;

示例性的, 本次计算所得到的最后一层数据层计算所得的参数和前一次计算所得到的最后一层数据层计算所得的参数均代入式 (3) 或式 (4) 的代价函数进行计算, 如果两个计算结果之差不大于预设的门限值, 则可以认为本次计算所得到的最后一层数据层计算所得的参数为最优参数, 如果大于预设的门限值, 则将本次计算所得到的最后一层数据层计算所得的参数是不收敛的, 并将本次计算所得到的最后一层数据层计算所得的参数作为下次计算的第一层数据层的参数初值, 继续计算推荐系统模型的参数。

S203: 根据最优参数与推荐系统模型获取每个用户对每个产品的评分预测值, 并根据评分预测值向用户推荐产品。

示例性的, 本实施例中, 将获得的最优参数带入式 (1) 或式 (2), 则可以获得每个用户对每个产品的评分预测值, 可以通过将相同用户对所有产品的评分预测值进行排列, 选择评分预测值最高的预设数量的产品推荐给用户。

S204: 产品推荐之后, 用户对产品做出的新的评分数据输入到推荐系统中, 以使得推荐系统能够根据实时的更新推荐系统模型的参数, 来保证高精度和高效率推荐的实时性。

本实施例提供了一种推荐方法, 通过并行计算提高了海量数据

环境下推荐系统的推荐效率，并且通过考虑用户隐式反馈提高了推荐系统的推荐效果。

本发明实施例提供了一种推荐设备 60，如图 6 所示，包括：

数据放置单元 601，用于将评分数据集中的评分数据分别放置到至少两个数据层，其中，评分数据与用户以及产品分别一一对应，且每一个数据层中的任意两个评分数据对应的用户以及产品均不相同；

并行计算单元 602，用于依据预设的推荐系统模型以及数据层中的评分数据，并行计算数据层中推荐系统模型的参数，并将每一层数据层的参数作为对应的下一层数据层的初值，直至获取推荐系统模型的最优参数；其中，推荐系统模型为每个用户对每个产品的评分预测值与平均分和所述推荐系统模型的参数之间的对应关系；

预测推荐单元 603，用于根据最优参数与推荐系统模型获取每个用户对每个产品的评分预测值，并根据评分预测值向用户推荐产品。

示例性的，评分数据集可以通过获取用户对产品的评分数据得到，也可以通过用户的浏览、购买记录信息得到，不仅可以得到用户对产品的显示反馈，也可以获取到用户偏好的隐式反馈，本发明实施例对此不做任何限制，优选的，评分数据集可以通过用户对产品的评分数据获得，本领域技术人员可以理解的，用户对产品的评分数据不仅显示的反馈了用户对产品的评价，而且也通过用户对产品评分的行为隐式的反馈了用户对产品的偏好，优选的，在本实施例中，评分数据集可以通过矩阵形式表示，其中，矩阵的不同行表示不同用户，矩阵的不同列表示不同产品；

进一步的，评分数据与用户以及产品分别一一对应，本领域技术人员可以理解的，由于评分数据所放置的数据层可以通过矩阵的形式进行表示，同时评分数据集也可以通过矩阵的形式进行表示，其中，矩阵的不同行表示不同用户，矩阵的不同列表示不同产品，于是可以将评分数据集中的评分数据分别放置到至少两个数据层矩

阵，并且所有数据层矩阵均与评分数据集的矩阵具有相同的行数和相同的列数，当每一个数据层中的任意两个评分数据对应的用户以及产品均不相同，即每一个数据层矩阵中的任意两个评分数据均不在同一行且均不在同一列的时候，就可以满足同一层数据层中的所有评分数据之间互相没有依赖关系，因此可以对同一层数据层中的评分数据进行并行计算，具体的放置步骤本发明实施例不作任何限定，任何能够使得每一个数据层矩阵中的任意两个评分数据均不在同一行且均不在同一列的放置方法均在本发明实施例的保护范围内；

进一步的，在本实施例中，数据放置单元 601 可以用于实现如图 3 所示放置方法，包括：

数据放置单元 601 选取评分数据集中的—个评分数据，将其放在第一层数据层矩阵中对应于该评分数据的用户和产品的位置上，此时，最后一层数 $l_{\max}=1$ ，其中，选取的方式本实施例不做限定，后续从评分数据集中选取数据的方式与第一次选取的方式相同，在此不作赘述；

数据放置单元 601 选取下一个评分数据，并从第一层数据层矩阵开始到最后一层对应的数据层矩阵，依次与其中的所有评分数据进行比较，是否满足评分数据在数据层矩阵中的对应位置所在的行和列均没有评分数据，若满足，则该评分数据放置于最先满足的数据层矩阵，若直至最后一层 $l_{\max}$ 对应的数据层矩阵也不满足，则最后一层数更新为 $l_{\max} \leftarrow l_{\max} + 1$ ，其中 $\leftarrow$ 符号表示将符号左边的数据更新为符号右边的数据，下同，并且将该评分数据放置于更新后的最后一层数据层；

数据放置单元 601 对评分数据集中剩余的评分数据依次重复上述放置评分数据的过程直至评分数据集中所有的评分数据均放置完毕。

例如，评分数据如矩阵 A 所示：

$$A = \begin{bmatrix} 3 & * & * & 2 & * & * \\ * & 2 & * & 2 & 1 & * \\ 4 & * & 5 & * & 5 & 2 \\ * & * & 2 & * & 3 & * \\ * & 1 & * & * & * & 5 \\ 1 & 2 & * & 3 & * & * \\ * & * & 3 & 5 & 2 & * \end{bmatrix},$$

数据放置单元 601 可以通过如图 3A 所示的放置方法, 示例性的, 具体操作过程如下: 评分数据集中选出一个评分数据 r_{ui} , 将其放入第一层, 然后再选择第二个评分数据 $r_{u'i'}$, 判断 u' 和 u 是否为同一个用户以及 i' 和 i 是否为同一个产品, 当 $u' \neq u$ 且 $i' \neq i$ 时, 该评分数据 $r_{u'i'}$ 放入第一层, 否则将其放入第二层。随后再取出第三个评分数据 $r_{u''i''}$, 如果该评分数据所对应的用户 u'' 和产品 i'' 与放置在第一层的用户和产品均不相同时, 则将其放入第一层, 否则与第二层的评分数据对应的用户和产品进行比较, 当满足 $u'' \neq u_{layer2}$ 且 $i'' \neq i_{layer2}$ 时, 则将其放入第二层, 否则放入第三层, 这里 u_{layer2} 表示第二层的用户, i_{layer2} 表示第二层的产品。依此类推, 直到集合 κ 的所有评分数据均放置完毕。

可以依次得到 5 个数据层, 从第一层数据层到最后一层也就是第五层数据层依次为:

$$\begin{aligned} (u_1, i_1, 3) & (u_2, i_2, 2) & (u_3, i_3, 5) & (u_4, i_5, 3) & (u_5, i_6, 5) & (u_6, i_4, 3) & l_1(\text{Layer } 1) \\ (u_1, i_4, 2) & (u_2, i_5, 1) & (u_3, i_1, 4) & (u_4, i_3, 2) & (u_5, i_2, 1) & & l_2(\text{Layer } 2) \\ (u_2, i_4, 2) & (u_3, i_5, 5) & (u_6, i_1, 1) & (u_7, i_3, 3) & & & l_3(\text{Layer } 3) \\ (u_3, i_6, 2) & (u_6, i_2, 2) & (u_7, i_4, 5) & & & & l_4(\text{Layer } 4) \\ (u_7, i_5, 2) & & & & & & l_5(\text{Layer } 5) \end{aligned}$$

通过矩阵形式表示可以是:

$$l_1 = \begin{bmatrix} 3 & * & * & * & * & * \\ * & 2 & * & * & * & * \\ * & * & 5 & * & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 5 \\ * & * & * & 3 & * & * \\ * & * & * & * & * & * \end{bmatrix}, l_2 = \begin{bmatrix} * & * & * & 2 & * & * \\ * & * & * & * & 1 & * \\ 4 & * & * & * & * & * \\ * & * & 2 & * & * & * \\ * & 1 & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}, l_3 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & 2 & * & * \\ * & * & * & * & 5 & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ 1 & * & * & * & * & * \\ * & * & 3 & * & * & * \end{bmatrix},$$

$$l_4 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & 2 \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & 2 & * & * & * & * \\ * & * & * & 5 & * & * \end{bmatrix}, \quad l_5 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & 2 & * \end{bmatrix}$$

其中，矩阵中出现的*表示该位置对应的用户没有对该位置对应的产品进行评分，并且 $A = l_1 + l_2 + l_3 + l_4 + l_5$ 。

进一步的，在本实施例中，数据放置单元 601 也可以用于实现如图 3B 所示放置方法，包括：

数据放置单元 601 选取评分数据集中的一个评分数据，将其放在第一层数据层中对应于该评分数据的用户和产品的位置上，此时，最后一层数 $l_{\max} = 1$ ，

数据放置单元 601 从评分数据集中选取的第二个评分数据开始，将该评分数据依次从最后一层数对应的数据层直至第一层数据层进行比较，直至能够从最后一层数到第一层数据层中找出最后一个满足该评分数据在数据层中对应位置的所在行和所在列上没有其他评分数据，则该评分数据放置于最后一个满足的数据层的对应位置，若从最后一层数到第一层均没有满足的数据评分矩阵，则最后一层数更新为 $l_{\max} \leftarrow l_{\max} + 1$ ，并且将该评分数据放置于更新后的最后一层数对应的数据层；

数据放置单元 601 对评分数据集中剩余的评分数据依次重复上述对第二个评分数据的放置过程直至评分数据集中所有的评分数据均放置完毕。

例如，仍以矩阵 A 为例，通过如图 3B 所示的放置方法，示例性的，具体操作过程如下：

从评分数据集中选出一个评分数据 r_{ui} ，将其放入第一层，然后再选择第二个评分数据 $r_{u'i'}$ ，判断 u' 和 u 是否为同一个用户以及 i' 和 i 是否为同一个产品，当 $u' \neq u$ 且 $i' \neq i$ 时，该评分数据 $r_{u'i'}$ 放入第一层，否则将其放入第

二层。随后再取出第三个评分数据 $r_{u''i''}$ ，将其与第二层已放置的评分数据进行比较(如果有第二层的话)，如果该评分数据所对应的用户 u'' 和产品 i'' 与放置在第二层的用户 u' 和产品 i' 均不相同，则继续将其与第一层已放置的评分数据进行比较，当与第一层已放置的评分数据也满足 $u'' \neq u$ 且 $i'' \neq i$ 时则将其放入第一层，否则放置第二层，而当该评分数据所对应的用户 u'' 和产品 i'' 与放置在第二层的用户 u' 和产品 i' 满足 $u'' = u'$ 或者 $i'' = i'$ 任意一个条件时，则直接将其放入第三层。依此类推，直到评分数据集中的所有评分数据均放置完毕，可以依次得到 7 个数据层，从第一层数据层到最后一层也就是第七层数据层依次为：

$$\begin{array}{lll}
 (u_1, i_1, 3) & (u_2, i_2, 2) & l_1(\text{Layer } 1) \\
 (u_1, i_4, 2) & (u_3, i_1, 4) & (u_5, i_2, 1) & l_2(\text{Layer } 2) \\
 (u_2, i_4, 2) & (u_3, i_3, 5) & (u_6, i_1, 1) & l_3(\text{Layer } 3) \\
 (u_2, i_5, 1) & (u_4, i_3, 2) & (u_6, i_2, 2) & l_4(\text{Layer } 4) \\
 (u_3, i_5, 5) & (u_6, i_4, 3) & (u_7, i_3, 3) & l_5(\text{Layer } 5) \\
 (u_3, i_6, 2) & (u_4, i_5, 3) & (u_7, i_4, 5) & l_6(\text{Layer } 6) \\
 (u_5, i_6, 5) & (u_7, i_5, 2) & & l_7(\text{Layer } 7)
 \end{array}$$

通过矩阵形式表示为：

$$\begin{array}{ll}
 l'_1 = \begin{bmatrix} 3 & * & * & * & * & * \\ * & 2 & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}, & l'_2 = \begin{bmatrix} * & * & * & 2 & * & * \\ * & * & * & * & * & * \\ 4 & * & * & * & * & * \\ * & * & * & * & * & * \\ * & 1 & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}, & l'_3 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & 2 & * & * \\ * & * & 5 & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ 1 & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}, \\
 l'_4 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & 1 & * \\ * & * & * & * & * & * \\ * & * & 2 & * & * & * \\ * & * & * & * & * & * \\ * & 2 & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}, & l'_5 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & 5 & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & 3 & * & * \\ * & * & 3 & * & * & * \end{bmatrix}, & l'_6 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & 2 \\ * & * & * & * & 3 & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & 5 & * & * \end{bmatrix},
 \end{array}$$

$$l_7' = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & 5 \\ * & * & * & * & * & * \\ * & * & * & * & 2 & * \end{bmatrix}$$

其中，*表示该位置对应的用户没有对该位置对应的产品进行评分，可以得到 $A = l_1' + l_2' + l_3' + l_4' + l_5' + l_6' + l_7'$ 。

示例性的，预设的系统模型可以是考虑用户隐式反馈的潜在因素模型，也可以是仅仅基于现实反馈的潜在因素模型，还可以包括考虑时空特性的推荐系统模型、非对称的潜在因素模型等，本发明实施例对此不做任何限定，

进一步的，在本实施例中，构建了一个改进的考虑用户隐式反馈的第一推荐系统模型，如式（1），以及一个未考虑用户隐式反馈的第二推荐系统模型，如式（2）；

在式（1）和式（2）中， $\hat{r}_{ui}$ 表示用户 u 对产品 i 的评分预测值， μ 表示评分数据集中的所有评分数据的平均值， b_u 表示用户 u 相对用户平均评分的偏移量， b_i 表示产品 i 相对产品平均评分的偏移量， q_i 表示产品因素矢量， T 表示转置运算符号， p_u 表示用户因素矢量；

进一步在式（1）中， $|N(u)|$ 表示用户 u 提供了隐式偏好的所有产品的集合大小， $N(u)$ 表示用户 u 提供了隐式偏好的所有产品的集合； $y_j \in f$ 表示与产品 j 相关联的因素矢量，其用于表征隐式反馈信息。而且 b_u ， b_i ， q_i ， p_u 和 y_j 为用户的推荐系统模型的未知参数；

示例性的，推荐设备 60 还可以包括代价函数生成单元 604，用于根据推荐系统模型得到的预测值与评分数据的均方误差和推荐系统模型的参数之间的关系得到推荐系统模型的代价函数，以及根据数据放置单元 601 得到的分层数据矩阵的评分数据，通过对上述模型有关的代价函数最优化问题的求解得到上述模型的未知参数，即并行计算数据层中推荐系统模型的参数，并将每一层数据层的参数

作为对应的下一层数据层的初值，直至获取推荐系统模型的最优参数，其中，最优参数就是上述模型的未知参数最优值，具体的，在本实施例中，第一推荐系统模型和第二推荐系统模型有关的代价函数可以分别表示为式(3)和式(4)，其中， $\|*\|^2$ 表示矢量*的所有元素的平方和， λ_1 与 λ_2 为正则化因子。

本领域技术人员可以理解的，并行计算单元 602 根据上述的模型对式(3)与式(4)进行最优化问题的求解，具体的求解过程具有相似性，不再赘述，具体的，以式(3)为例，如图 7 所示，并行计算单元 602 可以包括：

平均分计算子单元 6021，用于计算评分数据集中的所有评分数据的平均分；

分层计算子单元 6022，依次采用并行计算的方式计算每一层数据层的参数，并将每一层数据层计算所得的参数作为下一层数据层的参数初值；

示例性的，本实施例中，在第一次计算之前，可以将第一层数据层的参数 b_u ， b_i ， q_i ， p_u 和 y_j 以及 λ_1 与 λ_2 的初值进行随机设置，为了简便起见，参数 b_u ， b_i ， q_i ， p_u 和 y_j 的初值可以设置为标量参数为 0，矢量参数为向量 $\vec{0}$ ，其中，数字 0 上方的箭头符号为向量符号，而 λ_1 与 λ_2 可以任意设置为一个为比较小的正值，本发明实施例对此不作任何限定，用来表示正则化因子；

示例性的，从第一层数据层开始，依次的并行计算该层数据层的参数 b_u ， b_i ， q_i ， p_u 和 y_j 的值，并将计算所得的参数值作为下一层数据层的参数 b_u ， b_i ， q_i ， p_u 和 y_j 的初值，直至计算完最后一层数据层的参数 b_u ， b_i ， q_i ， p_u 和 y_j ，得到最后一层数据层计算所得的参数；具体的计算过程如下：

分层计算子单元 6022 根据第一层数据层的参数初值对第一层数据层所包含的评分数据对应的未知参数 b_u ， b_i ， q_i ， p_u 和 y_j 进行更新计算，由于同一层矩阵的评分数据之间没有相互依赖的关系，因此同一层矩阵的不同评分数据对应的未知参数 b_u ， b_i ， q_i ， p_u 和 y_j 的更新计算可以并行进行，并将更新后的参数作为下一层数据层的参

数初值对下一层数据层所包含的评分数据对应的参数进行更新计算，直至最后一层数据层所包含的评分数据对应的参数进行更新计算完毕，得到最后一层数据层计算所得的参数；

优选的，在本实施例中，可以通过梯度下降法进行更新计算，由于所有评分数据对应的参数的更新方法均相同，本领域技术人员可以理解，通过对一个评分数据 r_{ui} 对应的参数 b_u , b_i , q_i , p_u 和 y_j 进行更新计算步骤进行描述后，可以无需创造性的将计算步骤应用在其他评分数据中，分层计算子单元 6022 对式 (1) 的计算步骤具体可以如下：

首先，根据每一层数据层的参数初值和式 (1) 所示的推荐系统模型得到评分数据 r_{ui} 的初始估计值 $\hat{r}_{ui}$ ，进而根据该层数据层的评分数据和初始估计值 $\hat{r}_{ui}$ 得到该层数据层的评分误差 e_{ui} ；

接着，通过评分误差 e_{ui} 和式 (5) - 式 (9) 得到负梯度方向的参数的更新值，本领域技术人员可以理解的，式 (5) - 式 (8) 均可以根据相应的计算式进行参数的更新，计算过程及各式中出现的符号与参数的意义不再赘述，相应的，对式 (4) 的最优化问题的求解只需要进行式 (5) - 式 (8) 的更新计算，得到参数 b_u , b_i , q_i , p_u 即可；

但是对于式 (9) 的计算还需要结合评分数据集中的评分数据依次对 y_j 进行串行计算，降低了运算效率，因此本发明实施例提供了一种针对式 (9) 的计算方法，具体实施方式为：

当同一层矩阵有多个 y_j 需要计算时，利用评分数据集中 y_j 所表示的产品 j 对应的用户来并行计算其对应的梯度 Δy_j ，比如在评分数据集中，用户 1, 用户 2, 和用户 6 对产品 4 ($j=4$) 进行了评分，则可以并行计算用户 1, 用户 2, 和用户 6 对应的产品 j 的梯度 $\Delta y_j^{(1)}$, $\Delta y_j^{(2)}$, $\Delta y_j^{(6)}$ ，

其中， $\Delta y_j^{(1)} \leftarrow e_{11} \cdot 2^{-\frac{1}{2}} \cdot q_1 - \lambda_2 y_j^0$, $\Delta y_j^{(2)} \leftarrow e_{22} \cdot 3^{-\frac{1}{2}} \cdot q_2 - \lambda_2 y_j^0$, $\Delta y_j^{(6)} \leftarrow e_{64} \cdot 3^{-\frac{1}{2}} \cdot q_4 - \lambda_2 y_j^0$ ；

然后聚合得到该层 y_j 的值，即 $y_j \leftarrow y_j^0 + \gamma_2 (\Delta y_j^{(1)} + \Delta y_j^{(2)} + \Delta y_j^{(6)})$ ，其中， y_j^0 为 y_j 的初值。

此外，本发明实施例提供了另一种针对式 (8) 和式 (9) 的等

效计算方法，具体实施方式为：

将第一推荐模型中的表达式 $p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ 作为一个等效参数，并用辅助变量 z_u 来表示所述等效参数，即 $z_u = p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ ，其中辅助变量 z_u 与表达式 $p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ 等价，此时，第一推荐模型等价于 $\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot z_u$ ，其参数为 b_u, b_i, q_i, z_u ；

然后根据辅助变量 z_u 的梯度 $\Delta z_{ui} = 2e_{ui} \cdot q_i - \lambda_2 z_u$ ，并由此得到辅助变量 z_u 的更新公式 $z_u \leftarrow z_u^0 + \gamma_2 \cdot (2e_{ui} \cdot q_i - \lambda_2 z_u^0)$ ，其中 z_u^0 为 z_u 在该层计算时的初值。

通过本方法，可以只通过对辅助变量 z_u 的更新过程来替代对参数 p_u 和 y_j 的更新过程，此外，由于引入了辅助变量 z_u ，式 (8) 的关于参数 q_i 的更新也可以做出相应的变动，即 q_i 的更新公式变为 $q_i \leftarrow q_i^0 + \gamma_2 \cdot (e_{ui} \cdot z_u - \lambda_2 q_i^0)$ ，其中 q_i^0 为 q_i 在该层计算时的初值。

由此可知，引入辅助变量 z_u 一方面简化了计算，另一方面由于消除了 y_j 的计算，推荐模型的参数求解减少了一层内循环，从而在确保同样精度的前提下极大地提升了运算速度。

然后，可以得到第一层数据层的参数的更新值，接下来将此更新值作为第二层数据层的参数的初值，并且根据上述方法计算第二层数据层的参数的更新值，以此类推，得到最后一层数据层的参数的更新值。

收敛判断子单元 6023，用于根据最后一层数据层计算所得的参数判断推荐系统模型是否收敛，若收敛，则计算结束，得到最优参数；得到最优参数，若不收敛，则将最后一层数据层计算所得的参数作为第一层数据层的参数初值，继续通过分层计算子单元 6022 继续进行下一次分层计算。

优选的，收敛判断子单元 6023 可以将本次计算所得到的最后一层数据层计算所得的参数和前一次计算所得到的最后一层数据层计算所得的参数均代入式 (3) 或式 (4) 的代价函数进行计算，如果两个计算结果之差不大于预设的门限值，则可以认为本次计算所得到的最后一层数据层计算所得的参数为最优参数，如果大于预设的

门限值，则将本次计算所得到的最后一层数据层计算所得的参数是不收敛的，并将本次计算所得到的最后一层数据层计算所得的参数作为分层计算子单元 6022 继续进行下一次分层计算的第一层数据层的参数初值，继续计算推荐系统模型的参数。

示例性的，预测推荐单元 603 将获得的最优参数带入式 (1) 或式 (2)，则可以获得每个用户对每个产品的评分预测值，可以通过将相同用户对所有产品的评分预测值进行排列，选择评分预测值最高的预设数量的产品推荐给用户。

示例性的，如图 7 所示，推荐设备 60 还可以包括：

反馈单元 605，用于产品推荐之后，将用户对产品做出的新的评分数据输入到推荐系统中，以使得推荐系统能够根据实时的更新推荐系统模型的参数，来保证高精度和高效率推荐的实时性。

本实施例提供了一种推荐设备 60，通过并行计算提高了海量数据环境下推荐系统的推荐效率，并且通过考虑用户隐式反馈提高了推荐系统的推荐效果。

本实施例提供了一种推荐设备 60，如图 8 所示，包括：至少一个处理器 801、存储器 802 和至少一个通信总线 803，用于实现这些装置之间的连接和相互通信，其中，

通信总线 803 可以是工业标准体系结构 (Industry Standard Architecture, 简称为 ISA) 总线、外部设备互连 (Peripheral Component, 简称为 PCI) 总线或扩展工业标准体系结构 (Extended Industry Standard Architecture, 简称为 EISA) 总线等。

该总线 803 可以分为地址总线、数据总线、控制总线等。为便于表示，图 8 中仅用一条粗线表示，但并不表示仅有一根总线或一种类型的总线。

存储器 802 用于存储可执行程序代码以及处理器 801 的处理结果，该程序代码包括计算机操作指令。存储器 802 可能包含高速 RAM 存储器，也可能还包括非易失性存储器 (non-volatile memory)，

例如至少一个磁盘存储器。

处理器 801 可能是一个中央处理器 (Central Processing Unit, 简称为 CPU), 或者是特定集成电路 (Application Specific Integrated Circuit, 简称为 ASIC), 或者是被配置成实施本发明实施例的一个或多个集成电路。

处理器 801 用于执行存储器 704 中存储的可执行程序代码, 例如计算机程序来运行与可执行代码对应的程序。

处理器 801 用于:

将评分数据集中的评分数据分别放置到至少一个数据层, 其中, 评分数据与用户以及产品分别一一对应, 且每一个数据层中的任意两个评分数据对应的用户以及产品均不相同;

以及依据预设的推荐系统模型以及数据层中的评分数据, 并行计算数据层中推荐系统模型的参数, 并将每一层数据层的参数作为对应的下一层数据层的初值, 直至获取推荐系统模型的最优参数; 其中, 推荐系统模型为每个用户对每个产品的评分预测值与平均分和所述推荐系统模型的参数之间的对应关系;

以及根据最优参数与推荐系统模型获取每个用户对每个产品的评分预测值, 并根据评分预测值向用户推荐产品。

示例性的, 评分数据集可以通过获取用户对产品的评分数据得到, 也可以通过用户的浏览、购买记录信息得到, 不仅可以得到用户对产品的显示反馈, 也可以获取到用户偏好的隐式反馈, 本发明实施例对此不做任何限制, 优选的, 评分数据集可以通过用户对产品的评分数据获得, 本领域技术人员可以理解的, 用户对产品的评分数据不仅显示的反馈了用户对产品的评价, 而且也通过用户对产品评分的行为隐式的反馈了用户对产品的偏好, 优选的, 在本实施例中, 评分数据集可以通过矩阵形式表示, 其中, 矩阵的不同行表示不同用户, 矩阵的不同列表示不同产品;

进一步的, 评分数据与用户以及产品分别一一对应, 本领域技术人员可以理解的, 由于评分数据所放置的数据层可以通过矩阵的形式进行表示, 同时评分数据集也可以通过矩阵的形式进行表示,

其中，矩阵的不同行表示不同用户，矩阵的不同列表示不同产品，于是处理器 801 可以将评分数据集中的评分数据分别放置到至少两个数据层矩阵，并且所有数据层矩阵均与评分数据集的矩阵具有相同的行数和相同的列数，当每一个数据层中的任意两个评分数据对应的用户以及产品均不相同，即每一个数据层矩阵中的任意两个评分数据均不在同一行且均不在同一列的时候，就可以满足同一层数据层中的所有评分数据之间互相没有依赖关系，因此可以对同一层数据层中的评分数据进行并行计算，具体的放置步骤本发明实施例不作任何限定，任何能够使得每一个数据层矩阵中的任意两个评分数据均不在同一行且均不在同一列的放置方法均在本发明实施例的保护范围内；

可选的，在本实施例中，处理器 801 可以用于实现如图 3A 所示放置方法，包括：

处理器 801 选取评分数据集中的—个评分数据，将其放在第一层数据层矩阵中对应于该评分数据的用户和产品的位置上，此时，最后一层数设置为 $l_{\max}=1$ ，其中，选取的方式本实施例不做限定，后续从评分数据集中选取数据的方式与第一次选取的方式相同，在此不作赘述；

处理器 801 选取下一个评分数据，并从第一层数据层矩阵开始到最后一层对应的数据层矩阵，依次与其中的所有评分数据进行比较，是否满足评分数据在数据层矩阵中的对应位置所在的行和列均没有评分数据，若满足，则该评分数据放置于最先满足的数据层矩阵，若直至最后一层 $l_{\max}$ 对应的数据层也不满足，则最后一层数更新为 $l_{\max} \leftarrow l_{\max} + 1$ ，其中 $\leftarrow$ 符号表示将符号左边的数据更新为符号右边的数据，下同，并且将该评分数据放置于更新后的最后一层数据层；

处理器 801 对评分数据集中的剩余评分数据依次重复上述的放置过程直至评分数据集中所有的评分数据均放置完毕。例如，评分数据如矩阵 A 所示：

$$A = \begin{bmatrix} 3 & * & * & 2 & * & * \\ * & 2 & * & 2 & 1 & * \\ 4 & * & 5 & * & 5 & 2 \\ * & * & 2 & * & 3 & * \\ * & 1 & * & * & * & 5 \\ 1 & 2 & * & 3 & * & * \\ * & * & 3 & 5 & 2 & * \end{bmatrix},$$

通过如图 3A 所示的放置方法，示例性的，具体操作过程如下：
评分数据集中选出一个评分数据 r_{ui} ，将其放入第一层，然后再选择第二个评分数据 $r_{u'i'}$ ，判断 u' 和 u 是否为同一个用户以及 i' 和 i 是否为同一个产品，当 $u' \neq u$ 且 $i' \neq i$ 时，该评分数据 $r_{u'i'}$ 放入第一层，否则将其放入第二层。随后再取出第三个评分数据 $r_{u''i''}$ ，如果该评分数据所对应的用户 u'' 和产品 i'' 与放置在第一层的用户和产品均不相同时，则将其放入第一层，否则与第二层的评分数据对应的用户和产品进行比较，当满足 $u'' \neq u_{layer2}$ 且 $i'' \neq i_{layer2}$ 时，则将其放入第二层，否则放入第三层，这里 u_{layer2} 表示第二层的用户， i_{layer2} 表示第二层的产品。依此类推，直到集合 κ 的所有评分数据均放置完毕。

可以依次得到 5 个数据层，从第一层数据层到最后一层也就是第五层数据层依次为：

$$\begin{aligned} (u_1, i_1, 3) & (u_2, i_2, 2) & (u_3, i_3, 5) & (u_4, i_5, 3) & (u_5, i_6, 5) & (u_6, i_4, 3) & l_1(\text{Layer } 1) \\ (u_1, i_4, 2) & (u_2, i_5, 1) & (u_3, i_1, 4) & (u_4, i_3, 2) & (u_5, i_2, 1) & & l_2(\text{Layer } 2) \\ (u_2, i_4, 2) & (u_3, i_5, 5) & (u_6, i_1, 1) & (u_7, i_3, 3) & & & l_3(\text{Layer } 3) \\ (u_3, i_6, 2) & (u_6, i_2, 2) & (u_7, i_4, 5) & & & & l_4(\text{Layer } 4) \\ (u_7, i_5, 2) & & & & & & l_5(\text{Layer } 5) \end{aligned}$$

通过矩阵形式表示可以是：

$$l_1 = \begin{bmatrix} 3 & * & * & * & * & * \\ * & 2 & * & * & * & * \\ * & * & 5 & * & * & * \\ * & * & * & * & 3 & * \\ * & * & * & * & * & 5 \\ * & * & * & 3 & * & * \\ * & * & * & * & * & * \end{bmatrix}, \quad l_2 = \begin{bmatrix} * & * & * & 2 & * & * \\ * & * & * & * & 1 & * \\ 4 & * & * & * & * & * \\ * & * & 2 & * & * & * \\ * & 1 & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}, \quad l_3 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & 2 & * & * \\ * & * & * & * & 5 & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ 1 & * & * & * & * & * \\ * & * & 3 & * & * & * \end{bmatrix},$$

$$l_4 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & 2 \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & 2 & * & * & * & * \\ * & * & * & 5 & * & * \end{bmatrix}, \quad l_5 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & 2 & * \end{bmatrix}$$

其中，矩阵中出现的*表示该位置对应的用户没有对该位置对应的产品进行评分，可以得到 $A = l_1 + l_2 + l_3 + l_4 + l_5$ 。

进一步的，在本实施例中，处理器 801 也可以用于实现如图 3B 所示放置方法，包括：

处理器 801 选取评分数据集中的一个评分数据，将其放在第一层数据层中对应于该评分数据的用户和产品的位置上，此时，最后一层数 $l_{\max} = 1$ ；

处理器 801 从评分数据集中选取的第二个评分数据开始，将该评分数据依次从最后一层数对应的数据层直至第一层数据层进行比较，直至能够从最后一层数到第一层数据层中找出最后一个满足该评分数据在数据层中对应位置的所在行和所在列上没有其他评分数据，则该评分数据放置于最后一个满足的数据层的对应位置，若从最后一层数到第一层均没有满足的数据评分矩阵，则最后一层数更新为 $l_{\max} \leftarrow l_{\max} + 1$ ，并且将该评分数据放置于更新后的最后一层数对应的数据层；

处理器 801 对评分数据集中的剩余数据依次重复上述对第二个评分数据的放置过程直至评分数据集中所有的评分数据均放置完毕。

例如，仍以矩阵 A 为例，通过如图 3B 所示的放置方法，示例性的，具体操作过程如下：

从评分数据集中选出一个评分数据 r_{ui} ，将其放入第一层，然后再选择第二个评分数据 $r_{u'i'}$ ，判断 u' 和 u 是否为同一个用户以及 i' 和 i 是否为同一个产品，当 $u' \neq u$ 且 $i' \neq i$ 时，该评分数据 $r_{u'i'}$ 放入第一层，否则将其放入第

二层。随后再取出第三个评分数据 $r_{u''i''}$ ，将其与第二层已放置的评分数据进行比较(如果有第二层的话)，如果该评分数据所对应的用户 u'' 和产品 i'' 与放置在第二层的用户 u' 和产品 i' 均不相同，则继续将其与第一层已放置的评分数据进行比较，当与第一层已放置的评分数据也满足 $u'' \neq u$ 且 $i'' \neq i$ 时则将其放入第一层，否则放置第二层，而当该评分数据所对应的用户 u'' 和产品 i'' 与放置在第二层的用户 u' 和产品 i' 满足 $u'' = u'$ 或者 $i'' = i'$ 任意一个条件时，则直接将其放入第三层。依此类推，直到评分数据集中的所有评分数据均放置完毕，可以依次得到 7 个数据层，从第一层数据层到最后一层也就是第七层数据层依次为：

$$\begin{array}{lll}
 (u_1, i_1, 3) & (u_2, i_2, 2) & l_1(\text{Layer } 1) \\
 (u_1, i_4, 2) & (u_3, i_1, 4) & (u_5, i_2, 1) & l_2(\text{Layer } 2) \\
 (u_2, i_4, 2) & (u_3, i_3, 5) & (u_6, i_1, 1) & l_3(\text{Layer } 3) \\
 (u_2, i_5, 1) & (u_4, i_3, 2) & (u_6, i_2, 2) & l_4(\text{Layer } 4) \\
 (u_3, i_5, 5) & (u_6, i_4, 3) & (u_7, i_3, 3) & l_5(\text{Layer } 5) \\
 (u_3, i_6, 2) & (u_4, i_5, 3) & (u_7, i_4, 5) & l_6(\text{Layer } 6) \\
 (u_5, i_6, 5) & (u_7, i_5, 2) & & l_7(\text{Layer } 7)
 \end{array}$$

通过矩阵形式表示为：

$$\begin{array}{lll}
 l'_1 = \begin{bmatrix} 3 & * & * & * & * & * \\ * & 2 & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}, & l'_2 = \begin{bmatrix} * & * & * & 2 & * & * \\ * & * & * & * & * & * \\ 4 & * & * & * & * & * \\ * & * & * & * & * & * \\ * & 1 & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}, & l'_3 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & 2 & * & * \\ * & * & 5 & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ 1 & * & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}, \\
 l'_4 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & 1 & * \\ * & * & * & * & * & * \\ * & * & 2 & * & * & * \\ * & * & * & * & * & * \\ * & 2 & * & * & * & * \\ * & * & * & * & * & * \end{bmatrix}, & l'_5 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & 5 & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & 3 & * & * \\ * & * & 3 & * & * & * \end{bmatrix}, & l'_6 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & 2 \\ * & * & * & * & 3 & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & 5 & * & * \end{bmatrix},
 \end{array}$$

$$l'_7 = \begin{bmatrix} * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & * \\ * & * & * & * & * & 5 \\ * & * & * & * & * & * \\ * & * & * & * & 2 & * \end{bmatrix}$$

其中，矩阵中出现的*表示该位置对应的用户没有对该位置对应的产品进行评分，可以得到 $A = l'_1 + l'_2 + l'_3 + l'_4 + l'_5 + l'_6 + l'_7$ 。

示例性的，预设的系统模型可以是考虑用户隐式反馈的潜在因素模型，也可以是仅仅基于现实反馈的潜在因素模型，还可以包括考虑时空特性的推荐系统模型、非对称的潜在因素模型等，本发明实施例对此不做任何限定，

进一步的，在本实施例中，处理器 801 构建了一个改进的考虑用户隐式反馈的第一推荐系统模型，如式（1），以及一个未考虑用户隐式反馈的第二推荐系统模型，如式（2），

式（1）和式（2）中， $\hat{r}_{ui}$ 表示用户 u 对产品 i 的评分预测值， μ 表示评分数据集中的所有评分数据的平均值， b_u 表示用户 u 相对用户平均评分的偏移量， b_i 表示产品 i 相对产品平均评分的偏移量， q_i 表示产品因素矢量， T 表示转置运算符， p_u 表示用户因素矢量；

进一步在式（1）中， $|N(u)|$ 表示用户 u 提供了隐式偏好的所有产品的集合大小， $N(u)$ 表示用户 u 提供了隐式偏好的所有产品的集合； $y_j \in f$ 表示与产品 j 相关联的因素矢量，其用于表征隐式反馈信息。而且 b_u ， b_i ， q_i ， p_u 和 y_j 为用户的推荐系统模型的未知参数；

示例性的，处理器 801 还可以根据推荐系统模型得到的预测值与评分数据的均方误差和推荐系统模型的参数之间的关系得到推荐系统模型的代价函数，以及分层数据矩阵的评分数据，通过对上述模型有关的代价函数最优化问题的求解得到上述模型的未知参数，即并行计算数据层中推荐系统模型的参数，并将每一层数据层的参数作为对应的下一层数据层的初值，直至获取推荐系统模型的最优参数，其中，最优参数就是上述模型的未知参数最优值，具体的，

在本实施例中，第一推荐系统模型和第二推荐系统模型有关的代价函数可以分别表示为式(3)和式(4)；其中， $\|*\|^2$ 表示矢量*的所有元素的平方和， λ_1 与 λ_2 为正则化因子。

示例性的，处理器801根据上述的模型以及分层数据矩阵的评分数据，可以通过对上述模型有关的代价函数最优化问题的求解得到上述模型的未知参数，即并行计算数据层中推荐系统模型的参数，并将每一层数据层的参数作为对应的下一层数据层的初值，直至获取推荐系统模型的最优参数，本领域技术人员可以理解的，对式(3)与式(4)进行最优化问题的求解，具体的求解过程具有相似性，不再赘述，具体的，以式(3)为例，如图7所示，处理器801进一步用于：

计算评分数据集中的所有评分数据的平均分；

以及依次采用并行计算的方式计算每一层数据层的参数，并将每一层数据层计算所得的参数作为下一层数据层的参数初值；

示例性的，本实施例中，在第一次计算之前，可以将第一层数据层的参数 b_u ， b_i ， q_i ， p_u 和 y_j 以及 λ_1 与 λ_2 的初值进行随机设置，为了简便起见，参数 b_u ， b_i ， q_i ， p_u 和 y_j 的初值可以设置为标量参数为0，矢量参数为向量 $\vec{0}$ ，其中，数字0上方的箭头符号为向量符号，而 λ_1 与 λ_2 可以任意设置为一个为比较小的正值，本发明实施例对此不作任何限定，用来表示正则化因子；

示例性的，处理器801从第一层数据层开始，依次的并行计算该层数据层的参数 b_u ， b_i ， q_i ， p_u 和 y_j 的值，并将计算所得的参数值作为下一层数据层的参数 b_u ， b_i ， q_i ， p_u 和 y_j 的初值，直至计算完最后一层数据层的参数 b_u ， b_i ， q_i ， p_u 和 y_j ，得到最后一层数据层计算所得的参数；具体的计算过程如下：

处理器801根据第一层数据层的参数初值对第一层数据层所包含的评分数据对应的未知参数 b_u ， b_i ， q_i ， p_u 和 y_j 进行更新计算，由于同一层矩阵的评分数据之间没有相互依赖的关系，因此同一层矩阵的不同评分数据对应的未知参数 b_u ， b_i ， q_i ， p_u 和 y_j 的更新计算可以并行进行，并将更新后的参数作为下一层数据层的参数初值对下

一层数据层所包含的评分数据对应的参数进行更新计算，直至最后一层数据层所包含的评分数据对应的参数进行更新计算完毕，得到最后一层数据层计算所得的参数；

优选的，在本实施例中，处理器 801 可以通过梯度下降法进行更新计算，由于所有评分数据对应的参数的更新方法均相同，本领域技术人员可以理解，通过对一个评分数据 r_{ui} 对应的参数 b_u ， b_i ， q_i ， p_u 和 y_j 进行更新计算步骤进行描述后，可以无需创造性的将计算步骤应用在其他评分数据中，处理器 801 对式 (1) 的计算步骤具体可以如下：

首先，处理器 801 根据每一层数据层的参数初值和式 (1) 所示的推荐系统模型得到评分数据 r_{ui} 的初始估计值 $\hat{r}_{ui}$ ，进而根据该层数据层的评分数据和初始估计值 $\hat{r}_{ui}$ 得到该层数据层的评分误差 e_{ui} ；

接着，处理器 801 通过评分误差 e_{ui} 和式 (5) - 式 (9) 得到负梯度方向的参数的更新值，本领域技术人员可以理解的，式 (5) - 式 (8) 均可以根据相应的计算式进行参数的更新，计算过程及各式中出现的符号与参数的意义不再赘述，相应的，对式 (4) 的最优化问题的求解只需要进行式 (5) - 式 (8) 的更新计算，得到参数 b_u ， b_i ， q_i ， p_u 即可；

但是对于式 (9) 的计算，处理器 801 还需要结合评分数据集中的评分数据依次对 y_j 进行串行计算，降低了运算效率，因此本发明实施例提供了一种针对式 (9) 的计算方法，具体实施方式为：

当同一层矩阵有多个 y_j 需要计算时，处理器 801 利用评分数据集中 y_j 所表示的产品 j 对应的用户来并行计算其对应的梯度 Δy_j ，比如在评分数据集中，用户 1，用户 2，和用户 6 对产品 4 ($j=4$) 进行了评分，则处理器 801 可以并行计算用户 1，用户 2，和用户 6 对应的产品 j 的梯度 $\Delta y_j^{(1)}$ ， $\Delta y_j^{(2)}$ ， $\Delta y_j^{(6)}$ ，其中， $\Delta y_j^{(1)} \leftarrow e_{11} \cdot 2^{\frac{1}{2}} \cdot q_1 - \lambda_2 y_j^0$ ， $\Delta y_j^{(2)} \leftarrow e_{22} \cdot 3^{\frac{1}{2}} \cdot q_2 - \lambda_2 y_j^0$ ， $\Delta y_j^{(6)} \leftarrow e_{64} \cdot 3^{\frac{1}{2}} \cdot q_4 - \lambda_2 y_j^0$ ；

然后处理器 801 聚合得到该层 y_j 的值，即 $y_j \leftarrow y_j^0 + \gamma_2 (\Delta y_j^{(1)} + \Delta y_j^{(2)} + \Delta y_j^{(6)})$ ，其中， y_j^0 为 y_j 的初值。

此外，本发明实施例提供了另一种针对式 (8) 和式 (9) 的等

效计算方法，具体实施方式为：

将第一推荐模型中的表达式 $p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ 作为一个等效参数，并用辅助变量 z_u 来表示所述等效参数，即 $z_u = p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ ，其中辅助变量 z_u 与表达式 $p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ 等价，此时，第一推荐模型等价于 $\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \cdot z_u$ ，其参数为 b_u, b_i, q_i, z_u ；

然后根据辅助变量 z_u 的梯度 $\Delta z_{ui} = 2e_{ui} \cdot q_i - \lambda_2 z_u$ ，并由此得到辅助变量 z_u 的更新公式 $z_u \leftarrow z_u^0 + \gamma_2 \cdot (2e_{ui} \cdot q_i - \lambda_2 z_u^0)$ ，其中 z_u^0 为 z_u 在该层计算时的初值。

通过本方法，可以只通过对辅助变量 z_u 的更新过程来替代对参数 p_u 和 y_j 的更新过程，此外，由于引入了辅助变量 z_u ，式（8）的关于参数 q_i 的更新也可以做出相应的变动，即 q_i 的更新公式变为 $q_i \leftarrow q_i^0 + \gamma_2 \cdot (e_{ui} \cdot z_u - \lambda_2 q_i^0)$ ，其中 q_i^0 为 q_i 在该层计算时的初值。

由此可知，引入辅助变量 z_u 一方面简化了计算，另一方面由于消除了 y_j 的计算，推荐模型的参数求解减少了一层内循环，从而在确保同样精度的前提下极大地提升了运算速度。

然后，处理器 801 可以将该层数据层计算所得的参数作为下一层数据层的参数初值，接下来处理器 801 将此更新值作为第二层数据层的参数的初值，并且根据上述方法计算第二层数据层的参数的更新值，以此类推，处理器 801 得到最后一层数据层的参数的更新值。

示例性的，处理器 801 进一步的用于根据最后一层数据层计算所得的参数判断推荐系统模型是否收敛，若收敛，则计算结束，得到最优参数；得到最优参数，若不收敛，则将最后一层数据层计算所得的参数作为第一层数据层的参数初值，继续计算推荐系统模型的参数

优选的，处理器 801 可以本次计算所得到的最后一层数据层计算所得的参数和前一次计算所得到的最后一层数据层计算所得的参数均代入式（3）或式（4）的代价函数进行计算，如果两个计算结果之差不大于预设的门限值，则可以认为本次计算所得到的最后一层数据层计算所得的参数为最优参数，如果大于预设的门限值，则将

本次计算所得到的最后一层数据层计算所得的参数是不收敛的，并将本次计算所得到的最后一层数据层计算所得的参数作为下次计算的第一层数据层的参数初值，继续计算推荐系统模型的参数。

示例性的，处理器 801 将获得的最优参数带入式 (1) 或式 (2)，则可以获得每个用户对每个产品的评分预测值，可以通过将相同用户对所有产品的评分预测值进行排列，选择评分预测值最高的预设数量的产品推荐给用户。

示例性的，处理器 801 还可以用于产品推荐之后，将用户对产品做出的新的评分数据输入到推荐系统中，以使得推荐系统能够根据实时的更新推荐系统模型的参数，来保证高精度和高效率推荐的实时性。

本实施例提供了一种推荐设备 60，通过并行计算提高了海量数据环境下推荐系统的推荐效率，并且通过考虑用户隐式反馈提高了推荐系统的推荐效果。

在本申请所提供的几个实施例中，应该理解到，所揭露的系统，装置和方法，可以通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的，例如，单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口，装置或单元的间接耦合或通信连接，可以是电性，机械或其它的形式。

作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际需要选择其中的部分或者全部单元来实现本实施例方案的目的。

另外，在本发明各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理包括，也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现，也可以采用硬件加软件功能单元的形式实现。

上述以软件功能单元的形式实现的集成的单元，可以存储在一个计算机可读取存储介质中。上述软件功能单元存储在一个存储介质中，包括若干指令用以使得一台计算机设备（可以是个人计算机，服务器，或者网络设备等）执行本发明各个实施例方法的部分步骤。而前述的存储介质包括：U 盘、移动硬盘、只读存储器（Read-Only Memory，简称 ROM）、随机存取存储器（Random Access Memory，简称 RAM）、磁碟或者光盘等各种可以存储程序代码的介质。

最后应说明的是：以上实施例仅用以说明本发明的技术方案，而非对其限制；尽管参照前述实施例对本发明进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本发明各实施例技术方案的精神和范围。

权利要求书

1、一种推荐方法，其特征在于，包括：

将评分数据集中的评分数据分别放置到至少两个数据层，其中，所述评分数据与用户以及产品分别一一对应，且每一个所述数据层中的任意两个所述评分数据对应的用户以及产品均不相同；

依据预设的推荐系统模型以及所述数据层中的评分数据，并行计算所述数据层中推荐系统模型的参数，并将每一层数据层的参数作为对应的下一层数据层的初值，直至获取所述推荐系统模型的最优参数；其中，所述推荐系统模型为每个用户对每个产品的评分预测值与所述平均分和所述推荐系统模型的参数之间的对应关系；

根据所述最优参数与所述推荐系统模型获取每个用户对每个产品的评分预测值，并根据所述评分预测值向所述用户推荐产品。

2、根据权利要求 1 所述的方法，其特征在于，所述推荐系统模型包括提供了隐式反馈的推荐系统模型、未提供隐式反馈的推荐系统模型、考虑时空特性的推荐系统模型和非对称的潜在因素的推荐系统模型。

3、根据权利要求 1 或 2 所述的方法，其特征在于，所述推荐系统模型包括：

第一推荐系统模型

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j)$$

或者，第二推荐系统模型

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

在所述第一推荐模型和所述第二推荐模型中， $\hat{r}_{ui}$ 表示用户 u 对产品 i 的评分预测值， μ 表示所述评分数据集中的所有评分数据的平均值， b_u 表示所述用户 u 相对用户平均评分的偏移量， b_i 表示所述产品 i 相对产品平均评分的偏移量， q_i 表示产品因素矢量， T 表示转置运算符号， p_u 表示用户因素矢量，

进一步的，在所述第一推荐模型中， $|N(u)|$ 表示用户 u 提供了隐式偏好的所有产品的集合大小， $N(u)$ 表示用户 u 提供了隐式偏好的所有产品的集合； $y_j \in \mathbb{R}^f$ 表示与产品 j 相关联的因素矢量，其用于表征隐式反馈信息。

4、根据权利要求 3 所述的方法，其特征在于，还包括：

根据所述评分预测值与所述评分数据的均方误差和所述推荐系统模型的参数之间的关系得到所述推荐系统模型的代价函数，其中所述代价函数包括：

第一代价函数

$$\sum_{(u,i) \in \kappa} [r_{ui} - \mu - b_u - b_i - q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j)]^2 + \lambda_1 (b_u^2 + b_i^2) + \lambda_2 (\|q_i\|^2 + \|p_u\|^2 + \sum_{j \in N(u)} \|y_j\|^2)$$

或者，第二代价函数

$$\sum_{(u,i) \in \kappa} [r_{ui} - \mu - b_u - b_i - q_i^T p_u]^2 + \lambda_1 (b_u^2 + b_i^2) + \lambda_2 (\|q_i\|^2 + \|p_u\|^2)$$

其中， $\|*\|^2$ 表示矢量*的所有元素的平方和， λ_1 与 λ_2 为正则化因子。

5、根据权利要求 1-4 任一项所述的方法，其特征在于，所述依据预设的推荐系统模型以及所述数据层中的评分数据，并行计算所述数据层中推荐系统模型的参数，并将每一层数据层的参数作为对应的下一层数据层的初值，直至获取所述推荐系统模型的最优参数，包括：

A：计算所述评分数据集中的所有所述评分数据的平均分；

B：依次采用并行计算的方式计算每一层所述数据层的参数，并将每一层所述数据层计算所得的参数作为下一层数据层的参数初值；其中，所述第一层所述数据层的参数初值由系统设置；

C：根据最后一层所述数据层计算所得的参数判断所述推荐系统模型是否收敛，若收敛，则计算结束，得到所述最优参数；若不收敛，则将最后一层所述数据层计算所得的参数作为第一层所述数据层的参数初值，重复所述步骤 B、C。

6、根据权利要求 5 所述的方法，其特征在于，所述步骤 B，包括：

B1：根据每一层所述数据层的参数初值和所述推荐系统模型得到所述该层数据层的评分数据的初始估计值，进而根据所述该层数据层的评分数据和所述初始估计值得到所述该层数据层的评分误差；

B2：根据所述评分误差获取所述该层数据层计算所得的参数；

B3：将所述该层数据层计算所得的参数作为下一层数据层的参数初值，根据步骤 B1 和 B2 得到下一层数据层计算所得的参数，直至得

到最后一层所述数据层计算所得的参数。

7、根据权利要求 4 或 6 所述的方法，其特征在于，所述根据最后一层所述数据层计算所得的参数判断所述推荐系统模型是否收敛，包括：

将本次计算所得到的最后一层所述数据层计算所得的参数和上一次计算所得到的最后一层所述数据层计算所得的参数均代入所述代价函数进行计算，若代入所述代价函数进行计算的结果之差不大于预设的门限值，则本次计算所得到的最后一层所述数据层计算所得的参数是收敛的，否则，本次计算所得到的最后一层所述数据层计算所得的参数是不收敛的。

8、根据权利要求 6 所述的方法，其特征在于，所述根据所述评分误差获取所述该层数据层计算所得的参数，进一步包括：

将所述第一推荐模型中的表达式 $p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ 作为一个等效参数，并用辅助变量 z_u 来表示所述等效参数，即 $z_u = p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ ；

然后根据所述辅助变量 z_u 的梯度 $\Delta z_{ui} = 2e_{ui} \cdot q_i - \lambda_2 z_u$ ，获取所述辅助变量 z_u ，即得到所述等效参数；以及根据所述辅助变量 z_u 获取所述参数 q_i ，即 $q_i \leftarrow q_i + \gamma_2 (e_{ui} \cdot z_u - \lambda_2 q_i)$ ，其中， $\leftarrow$ 符号表示更新符号，即用更新符号右边的计算值替代更新符号左边的变量值，更新符号右边出现的参数均为相应的参数的初值，更新符号左边出现的参数均为参数的更新值。

9、一种推荐设备，其特征在于，包括：

数据放置单元，用于将评分数据集中的评分数据分别放置到至少两个数据层，其中，所述评分数据与用户以及产品分别一一对应，且每一个所述数据层中的任意两个所述评分数据对应的用户以及产品均不相同；

并行计算单元，用于依据预设的推荐系统模型以及所述数据层中的评分数据，并行计算所述数据层中推荐系统模型的参数，并将每一层数据层的参数作为对应的下一层数据层的初值，直至获取所述推荐系统模型的最优参数；其中，所述推荐系统模型为每个用户对每个产

品的评分预测值与所述平均分和所述推荐系统模型的参数之间的对应关系；

预测推荐单元，用于根据所述最优参数与所述推荐系统模型获取每个用户对每个产品的评分预测值，并根据所述评分预测值向所述用户推荐产品。

10、根据权利要求 9 所述的设备，其特征在于，所述推荐系统模型包括提供了隐式反馈的推荐系统模型、未提供隐式反馈的推荐系统模型、考虑时空特性的推荐系统模型和非对称的潜在因素的推荐系统模型。

11、根据权利要求 9 或 10 所述的设备，其特征在于，所述推荐系统模型包括：

所述推荐系统模型包括：

第一推荐系统模型

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j)$$

或者，第二推荐系统模型

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

在所述第一推荐模型和所述第二推荐模型中， $\hat{r}_{ui}$ 表示用户 u 对产品 i 的评分预测值， μ 表示所述评分数据集中的所有评分数据的平均值， b_u 表示所述用户 u 相对用户平均评分的偏移量， b_i 表示所述产品 i 相对产品平均评分的偏移量， q_i 表示产品因素矢量， T 表示转置运算符， p_u 表示用户因素矢量，

进一步的，在所述第一推荐模型中， $|N(u)|$ 表示用户 u 提供了隐式偏好的所有产品的集合大小， $N(u)$ 表示用户 u 提供了隐式偏好的所有产品的集合； $y_j \in \mathcal{F}$ 表示与产品 j 相关联的因素矢量，其用于表征隐式反馈信息。

12、根据权利要求 11 所述的设备，其特征在于，还包括：代价函数生成单元，用于根据所述评分预测值与所述评分数据的均方误差和所述推荐系统模型的参数之间的关系得到所述推荐系统模型的代价函数，其中所述代价函数包括：

第一代价函数

$$\sum_{(u,i) \in \kappa} [r_{ui} - \mu - b_u - b_i - q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j)]^2 + \lambda_1 (b_u^2 + b_i^2) + \lambda_2 (\|q_i\|^2 + \|p_u\|^2 + \sum_{j \in N(u)} \|y_j\|^2)$$

或者，第二代价函数

$$\sum_{(u,i) \in \kappa} [r_{ui} - \mu - b_u - b_i - q_i^T p_u]^2 + \lambda_1 (b_u^2 + b_i^2) + \lambda_2 (\|q_i\|^2 + \|p_u\|^2)$$

其中， $\|\cdot\|^2$ 表示矢量*的所有元素的平方和， λ_1 与 λ_2 为正则化因子。

13、根据权利要求 9-12 任一项所述的设备，其特征在于，所述并行计算单元，包括：

平均分计算子单元，用于计算所述评分数据集中的所有所述评分数据的平均分；

分层计算子单元，用于依次采用并行计算的方式计算每一层所述数据层的参数，并将每一层所述数据层计算所得的参数作为下一层数据层的参数初值；其中，所述第一层所述数据层的参数初值由系统设置；

收敛判断子单元，用于根据最后一层所述数据层计算所得的参数判断所述推荐系统模型是否收敛，若收敛，则计算结束，得到所述最优参数；若不收敛，则将最后一层所述数据层计算所得的参数作为第一层所述数据层的参数初值，并将所述参数初值传输至所述分层计算子单元重复进行分层计算。

14、根据权利要求 13 所述的设备，其特征在于，所述分层计算子单元进一步用于，

评分误差生成模块，用于根据每一层所述数据层的参数初值和所述推荐系统模型得到所述该层数据层的评分数据的初始估计值，进而根据所述该层数据层的评分数据和所述初始估计值得到所述该层数据层的评分误差；

参数计算模块，用于根据所述评分误差获取所述该层数据层计算所得的参数；

计算控制模块，用于将所述该层数据层计算所得的参数作为下一层数据层的参数初值，通过所述评分误差生成模块和所述参数计算模块得到下一层数据层计算所得的参数，直至得到最后一层所述数据层计算所得的参数。

15、根据权利要求 13 或 14 所述的设备，其特征在于，所述收敛判断子单元进一步用于，

将本次计算所得到的最后一层所述数据层计算所得的参数和上一次计算所得到的最后一层所述数据层计算所得的参数均代入所述代价函数进行计算，若代入所述代价函数进行计算的结果之差不大于预设的门限值，则本次计算所得到的最后一层所述数据层计算所得的参数是收敛的，否则，本次计算所得到的最后一层所述数据层计算所得的参数是不收敛的。

16、根据权利要求 14 所述的设备，其特征在于，所述参数计算模块进一步用于，

将所述第一推荐模型中的表达式 $p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ 作为一个等效参数，并用辅助变量 z_u 来表示所述等效参数，即 $z_u = p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ ；

然后根据所述辅助变量 z_u 的梯度 $\Delta z_{ui} = 2e_{ui} \cdot q_i - \lambda_2 z_u$ ，获取所述辅助变量 z_u ，即得到所述等效参数；以及根据所述辅助变量 z_u 获取所述参数 q_i ，即 $q_i \leftarrow q_i + \gamma_2 (e_{ui} \cdot z_u - \lambda_2 q_i)$ ，其中， $\leftarrow$ 符号表示更新符号，即用更新符号右边的计算值替代更新符号左边的变量值，更新符号右边出现的参数均为相应的参数的初值，更新符号左边出现的参数均为参数的更新值。

17、一种推荐设备，包括处理器和存储器，其中，

所述处理器用于，将评分数据集中的评分数据分别放置到至少两个数据层，其中，所述评分数据与用户以及产品分别一一对应，且每一个所述数据层中的任意两个所述评分数据对应的用户以及产品均不相同；

以及依据预设的推荐系统模型以及所述数据层中的评分数据，并行计算所述数据层中推荐系统模型的参数，并将每一层数据层的参数作为对应的下一层数据层的初值，直至获取所述推荐系统模型的最优参数；其中，所述推荐系统模型为每个用户对每个产品的评分预测值与所述平均分和所述推荐系统模型的参数之间的对应关系；

以及根据所述最优参数与所述推荐系统模型获取每个用户对每

个产品的评分预测值，并根据所述评分预测值向所述用户推荐产品；

所述存储器用于保存评分数据集以及处理器所执行的程序和执行的结果。

18、根据权利要求 17 所述的设备，其特征在于，所述推荐系统模型包括提供了隐式反馈的推荐系统模型、未提供隐式反馈的推荐系统模型、考虑时空特性的推荐系统模型和非对称的潜在因素的推荐系统模型。

19、根据权利要求 17 或 18 所述的设备，所述推荐系统模型包括：
所述推荐系统模型包括：

第一推荐系统模型

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j)$$

或者，第二推荐系统模型

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u$$

在所述第一推荐模型和所述第二推荐模型中， $\hat{r}_{ui}$ 表示用户 u 对产品 i 的评分预测值， μ 表示所述评分数据集中的所有评分数据的平均值， b_u 表示所述用户 u 相对用户平均评分的偏移量， b_i 表示所述产品 i 相对产品平均评分的偏移量， q_i 表示产品因素矢量， T 表示转置运算符号， p_u 表示用户因素矢量，

进一步的，在所述第一推荐模型中， $|N(u)|$ 表示用户 u 提供了隐式偏好的所有产品的集合大小， $N(u)$ 表示用户 u 提供了隐式偏好的所有产品的集合； $y_j \in \mathcal{F}$ 表示与产品 j 相关联的因素矢量，其用于表征隐式反馈信息。

20、根据权利要求 19 所述的设备，其特征在于，所述处理器还用于，根据所述评分预测值与所述评分数据的均方误差和所述推荐系统模型的参数之间的关系得到所述推荐系统模型的代价函数，其中所述代价函数包括：

第一代价函数

$$\begin{aligned} & \sum_{(u,i) \in \mathcal{K}} [r_{ui} - \mu - b_u - b_i - q_i^T (p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j)]^2 \\ & + \lambda_1 (b_u^2 + b_i^2) + \lambda_2 (\|q_i\|^2 + \|p_u\|^2 + \sum_{j \in N(u)} \|y_j\|^2) \end{aligned}$$

或者，第二代价函数

$$\sum_{(u,i) \in K} [r_{ui} - \mu - b_u - b_i - q_i^T p_u]^2 + \lambda_1 (b_u^2 + b_i^2) + \lambda_2 (\|q_i\|^2 + \|p_u\|^2)$$

其中， $\|*\|^2$ 表示矢量*的所有元素的平方和， λ_1 与 λ_2 为正则化因子。

21、根据权利要求 18-20 任一项的设备，其特征在于，所述处理器用于，

A：计算所述评分数据集中的所有所述评分数据的平均分；

B：依次采用并行计算的方式计算每一层所述数据层的参数，并将每一层所述数据层计算所得的参数作为下一层数据层的参数初值；其中，所述第一层所述数据层的参数初值由系统设置；

C：根据最后一层所述数据层计算所得的参数判断所述推荐系统模型是否收敛，若收敛，则计算结束，得到所述最优参数；若不收敛，则将最后一层所述数据层计算所得的参数作为第一层所述数据层的参数初值，重复所述步骤 B、C。

22、根据权利要求 21 所述的设备，其特征在于，所述处理器用于，

B1：根据每一层所述数据层的参数初值和所述推荐系统模型得到所述该层数据层的评分数据的初始估计值，进而根据所述该层数据层的评分数据和所述初始估计值得到所述该层数据层的评分误差；

B2：根据所述评分误差获取所述该层数据层计算所得的参数；

B3：将所述该层数据层计算所得的参数作为下一层数据层的参数初值，根据步骤 B1 和 B2 得到下一层数据层计算所得的参数，直至得到最后一层所述数据层计算所得的参数。

23、根据权利要求 21 或 22 所述的设备，其特征在于，所述处理器用于，

将本次计算所得到的最后一层所述数据层计算所得的参数和前一次计算所得到的最后一层所述数据层计算所得的参数均代入所述代价函数进行计算，若代入所述代价函数进行计算的结果之差不大于预设的门限值，则本次计算所得到的最后一层所述数据层计算所得的参数是收敛的，否则，本次计算所得到的最后一层所述数据层计算所得的参数是不收敛的。

24、根据权利要求 22 所述的设备，其特征在于，所述处理器用于根据所述评分误差获取所述该层数据层计算所得的参数，进一步包

括:

所述处理器将所述第一推荐模型中的表达式 $p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$ 作为一个等效参数, 并用辅助变量 z_u 来表示所述等效参数, 即 $z_u = p_u + |N(u)|^{-\frac{1}{2}} \sum_{j \in N(u)} y_j$;

然后根据所述辅助变量 z_u 的梯度 $\Delta z_{ui} = 2e_{ui} \cdot q_i - \lambda_2 z_u$, 获取所述辅助变量 z_u , 即得到所述等效参数; 以及根据所述辅助变量 z_u 获取所述参数 q_i , 即 $q_i \leftarrow q_i + \gamma_2 (e_{ui} \cdot z_u - \lambda_2 q_i)$, 其中, $\leftarrow$ 符号表示更新符号, 即用更新符号右边的计算值替代更新符号左边的变量值, 更新符号右边出现的参数均为相应的参数的初值, 更新符号左边出现的参数均为参数的更新值。

1/7

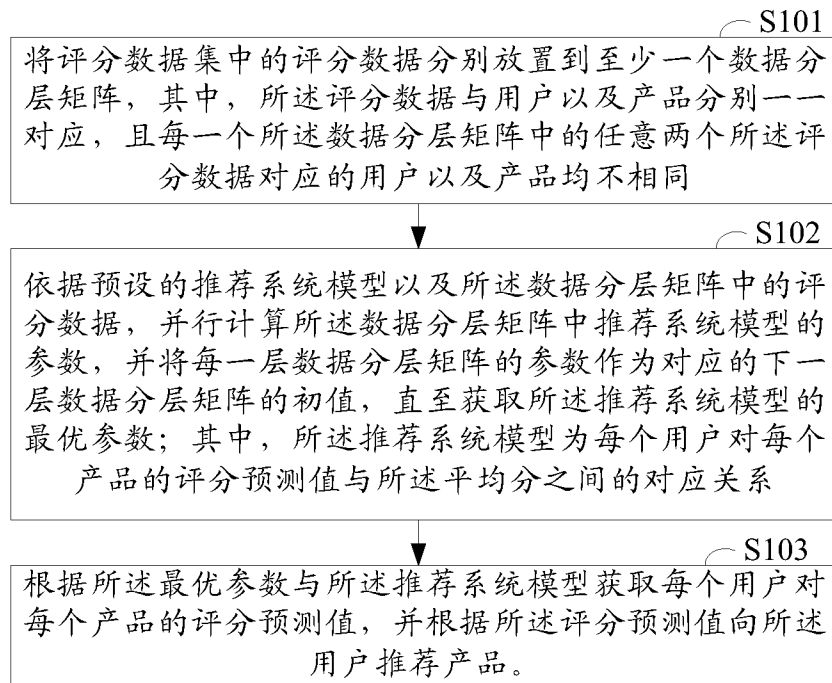


图 1

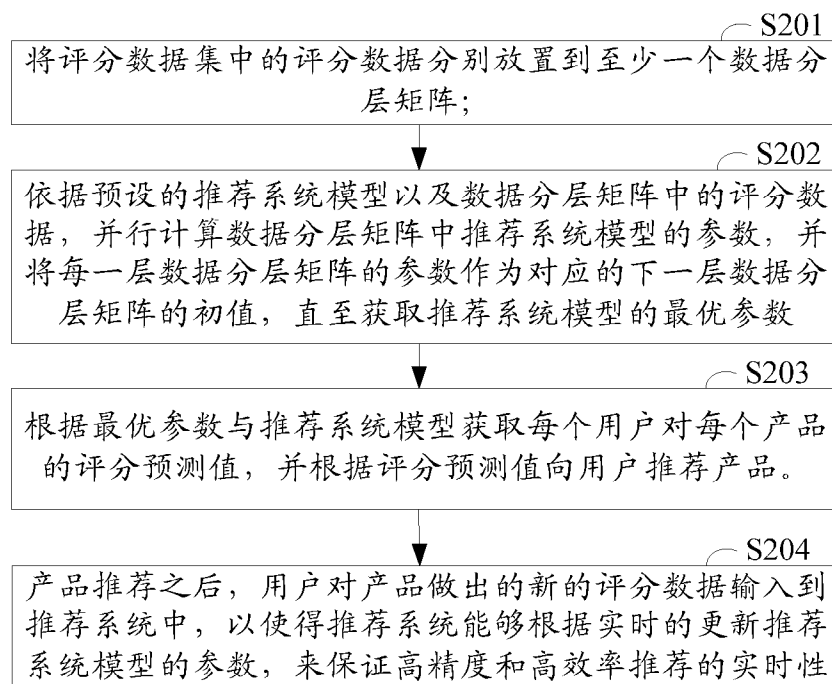


图 2

2/7

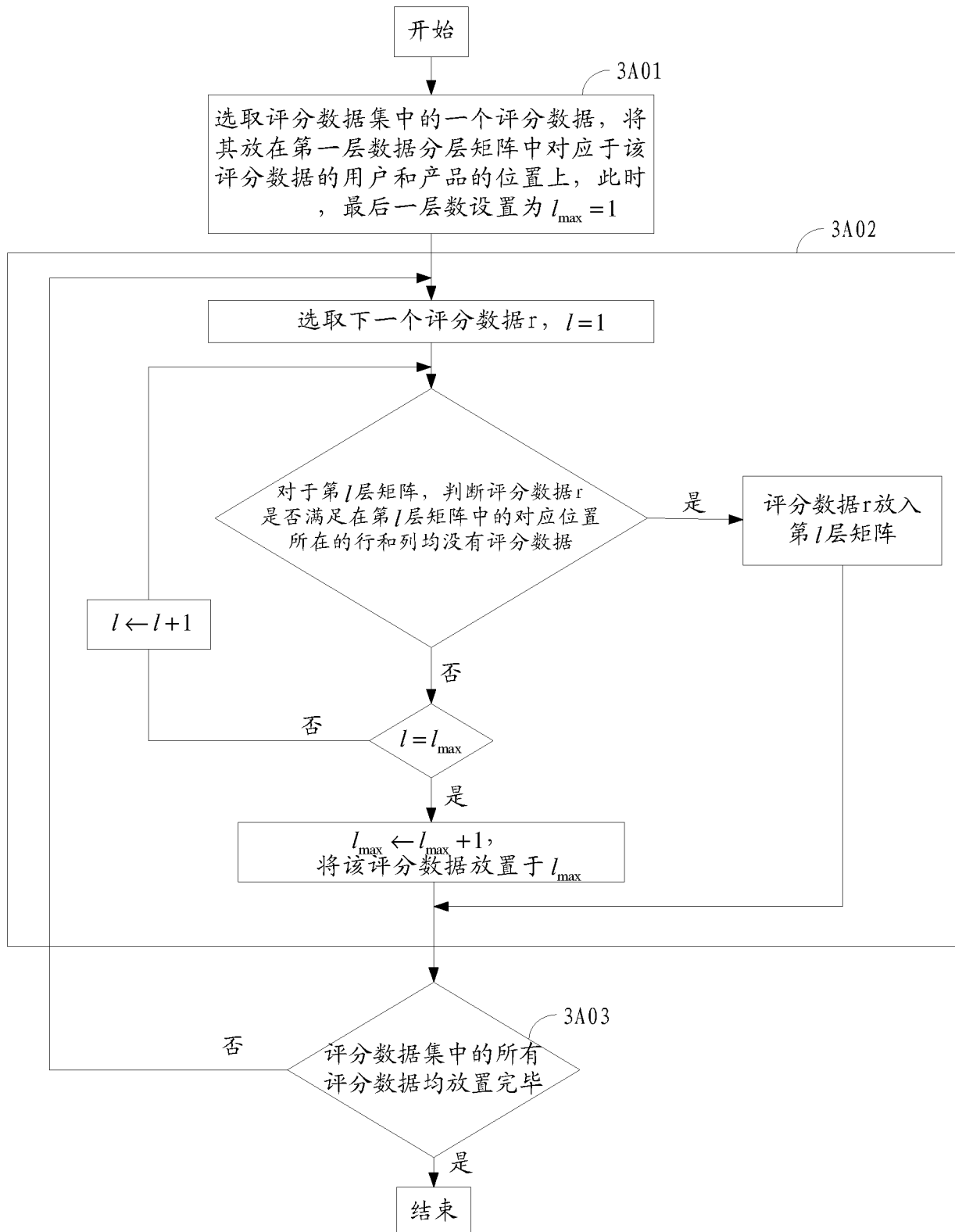


图 3A

3/7

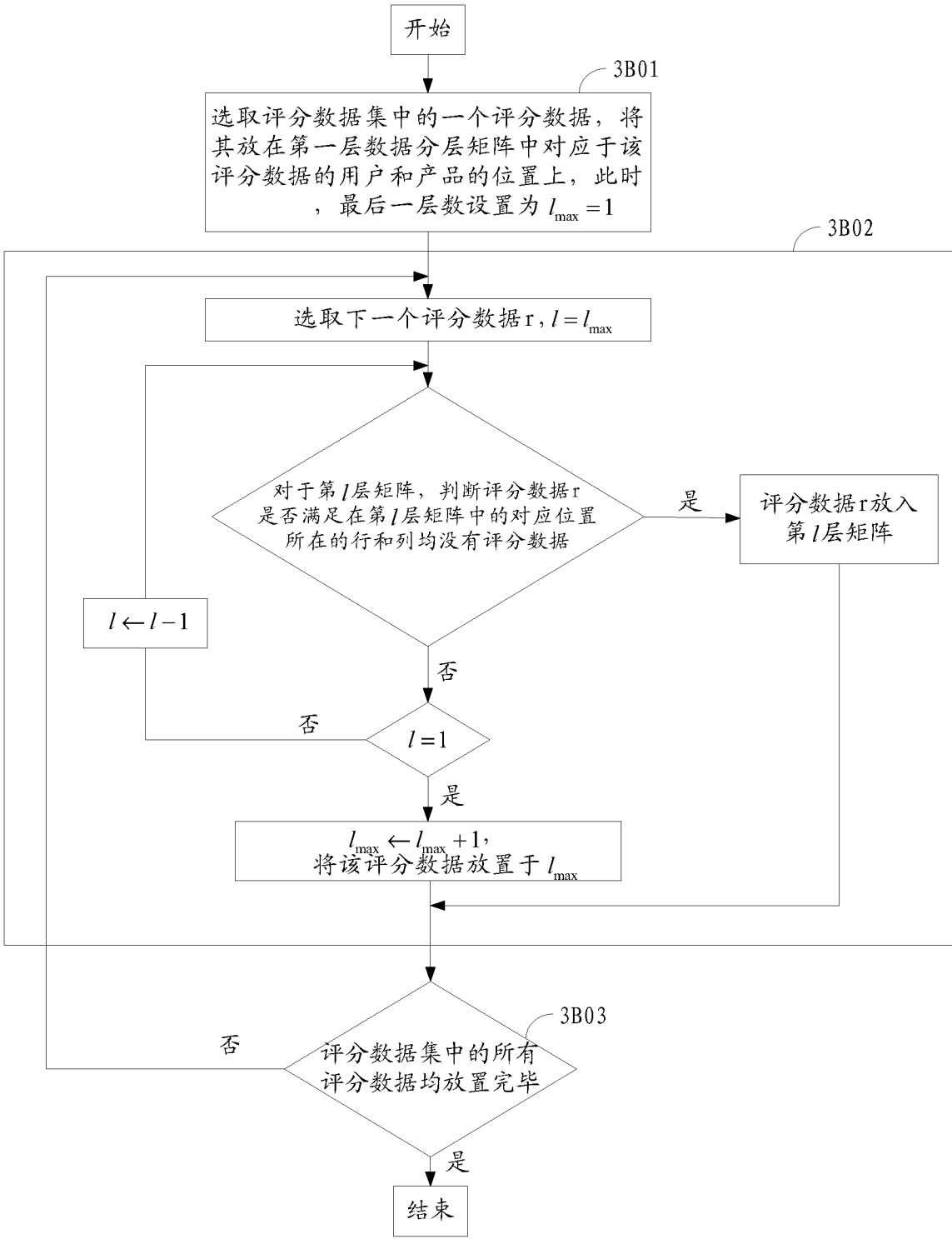


图 3B

图 3

5/7

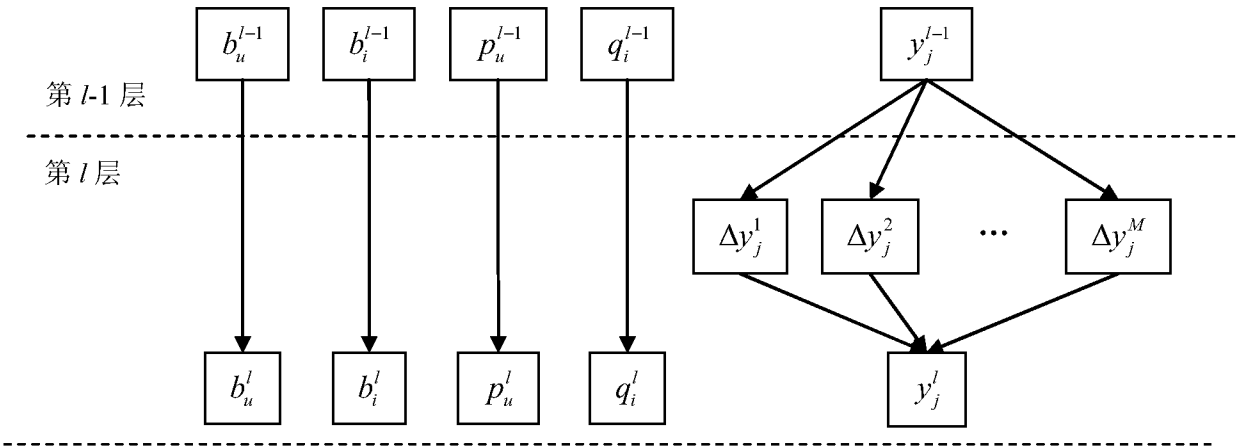


图 5A

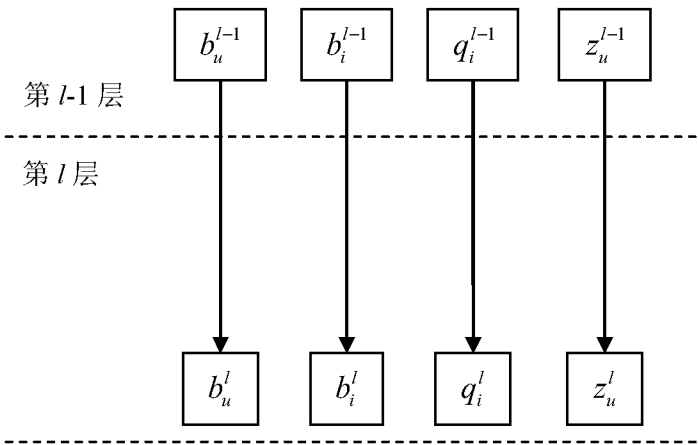


图 5B

图 5

6/7

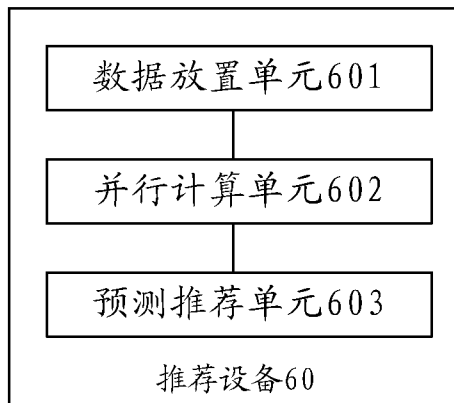


图 6

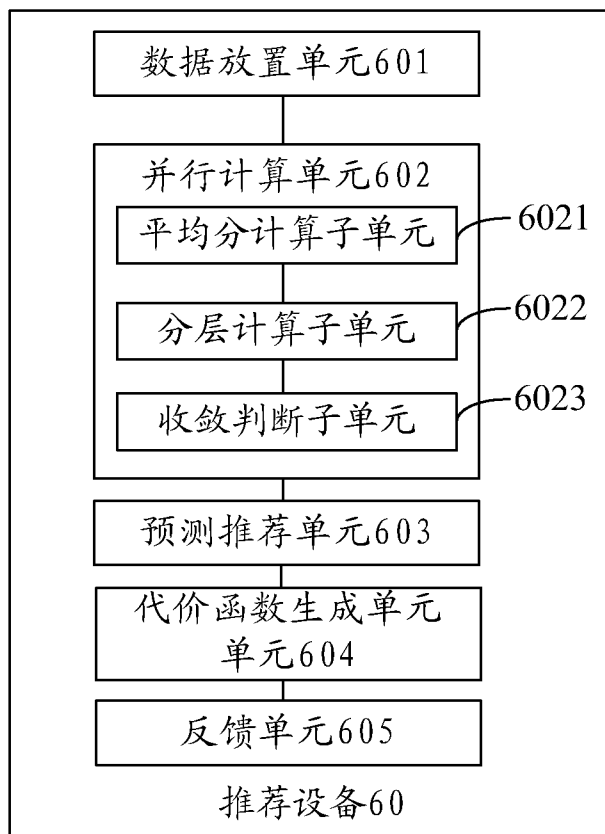


图 7

7/7

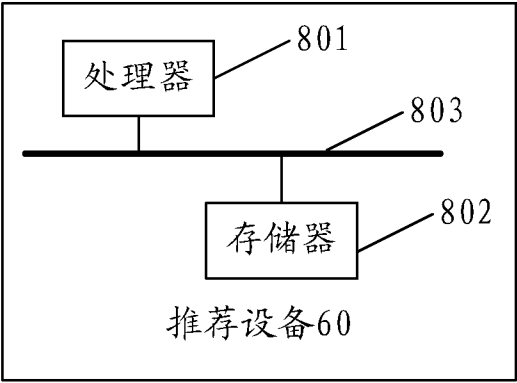


图 8

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2013/083218

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/30 (2006.01) i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F G06Q G06N H04W

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, CNKI, EPODOC, WPI: evaluate, preference, electronic commerce, value, grade, rank, recommend, commend, favorite, satisfaction, business, latent, predict

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 102262764 A (WANG, Xi), 30 November 2011 (30.11.2011), abstract, and description, paragraphs [0028]-[0040]	1-24
A	CN 101437220 A (GUANGZHOU WUDU INFORMATION TECHNOLOGY CO., LTD.), 20 May 2009 (20.05.2009), the whole document	1-24
A	US 2002065797 A1 (WIZSOFT LTD.), 30 May 2002 (30.05.2002), the whole document	1-24
A	US 2010185579 A1 (HONG, K.S. et al.), 22 July 2010 (22.07.2010), the whole document	1-24

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:	“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
“A” document defining the general state of the art which is not considered to be of particular relevance	“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
“E” earlier application or patent but published on or after the international filing date	“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	“&” document member of the same patent family
“O” document referring to an oral disclosure, use, exhibition or other means	
“P” document published prior to the international filing date but later than the priority date claimed	

Date of the actual completion of the international search 15 May 2014 (15.05.2014)	Date of mailing of the international search report 17 June 2014 (17.06.2014)
Name and mailing address of the ISA/CN: State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No.: (86-10) 62019451	Authorized officer ZHANG, Yingxin Telephone No.: (86-10) 62413366

International application No.
PCT/CN2013/083218

Form PCT/ISA/210 (patent family annex) (July 2009)

A. 主题的分类		
G06F 17/30(2006.01)i		
按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类		
B. 检索领域		
检索的最低限度文献(标明分类系统和分类号)		
G06F G06Q G06N H04W		
包含在检索领域中的除最低限度文献以外的检索文献		
在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))		
CNPAT, CNKI, EPODOC, WPI: 评分、分值、评价、推荐、偏好、偏爱、满意度、电子商务、隐含、预测、value, grade, rank, recommend, commend, favorite, satisfaction, business, latent, predict		
C. 相关文件		
类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
A	CN 102262764A (王希) 2011年 11月 30日 (2011 - 11 - 30) 摘要、说明书[0028-0040]段	1-24
A	CN 101437220A (广州五度信息技术有限公司) 2009年 5月 20日 (2009 - 05 - 20) 全文	1-24
A	US 2002065797A1 (WIZSOFT LTD.) 2002年 5月 30日 (2002 - 05 - 30) 全文	1-24
A	US 2010185579A1 (HONG, KWANG SEOK等) 2010年 7月 22日 (2010 - 07 - 22) 全文	1-24
☐ 其余文件在C栏的续页中列出。 ☒ 见同族专利附件。		
* 引用文件的具体类型: “A” 认为不特别相关的表示了现有技术一般状态的文件 “E” 在国际申请日的当天或之后公布的在先申请或专利 “L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的) “O” 涉及口头公开、使用、展览或其他方式公开的文件 “P” 公布日先于国际申请日但迟于所要求的优先权日的文件 “T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件 “X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性 “Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性 “&” 同族专利的文件		
国际检索实际完成的日期		国际检索报告邮寄日期
2014年 5月 15日		2014年 6月 17日
ISA/CN的名称和邮寄地址		授权官员
中华人民共和国国家知识产权局(ISA/CN) 北京市海淀区蓟门桥西土城路6号 100088 中国		张迎新
传真号 (86-10)62019451		电话号码 (86-10)62413366

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2013/083218

检索报告引用的专利文件		公布日 (年/月/日)	同族专利	公布日 (年/月/日)
CN	102262764A	2011年 11月 30日	无	
CN	101437220A	2009年 5月 20日	无	
US	2002065797A1	2002年 5月 30日	无	
US	2010185579A1	2010年 7月 22日	KR 20100086296A	2010年 7月 30日