社交网络中的个性化推荐系统

宁小东，*学号 2120151024*，黄建峰，*学号 2120150994*，王新灵，*学号 2120151042*

***摘要*—我们通过数据分析、数据清洗、数据合并以及逻辑回归的方式对 4.08GB 微博数据进行了处理，并预测待测数据集中用户可能接受的推荐 Item。在数据分析阶段，我们对原始数据集的各个部分进行了数据摘要和可视化，选取了其中部分数据作为预测的参考字段；然后，我们对数据进行了清洗，淘汰了部分不合理或输入非法的数据；以清洗后的数据集为基础，将数据分析中的参考字段进行数据合并，获取了适合预测的数据集；最终，我们使用逻辑回归算法预测了待测数据集的推荐 Item。预测结果显示，用户接受的 Item 其 ID 集中于一定范围内；存在最频繁的推荐 Item。在整个推荐过程中，我们的方法充分利用了原始数据，剔除了其中的噪声，并能够高效的预测推荐结果。**

***关键词*—数据挖掘，微博推荐系统，逻辑回归**

I. 简介

所要解决的问题是：取微博中的用户属性、SNS 社交关系、过去 30 天内的历史 Item 推荐记录等，预测接下来最有可能被用户接受的推荐 Item 列表。整个预测过程中涉及的各名词定义如下：

1) *Item*
   Item 是指微博中的特定用户，可能是个人、组织或集体，用于推荐给其他用户。例如，名人或知名组织可能会作为备选的“Item 集合”推荐给用户；
2) *推送（Tweet）*
   是指用户将信息上传到微博系统的行为，或者是指推文本身；
3) *评论*
   用户可以对推送（Tweet）进行评论。评论不会像推送或分享（Retweet）那样显示给他的关注着，而是出现在该推送的评论历史中；
4) *关注者（Follower）/被关注者（Followee）*
   如果用户 B 关注了用户 A，那么 B 就是 A 的关注者，A 是 B 的被关注者。

针对以上问题，我们计划通过 4 个步骤解决：数据分析，找出适合预测任务的参考字段；数据清洗，淘汰部分无效数据；数据合并，以清洗后的数据集为基础，按照数据分析的参考字段合并相关数据集；数据预测，通过逻辑回归的方式预测推荐 Item，完成微博个性化推荐系统的任务。

II. 问题陈述

我们从网上抓取了 4.08GB 的训练/测试数据集。这些数据分为 7 个部分：

1) *训练集*
   格式为：（用户 ID）\t（Item ID）\t（结果），其中结果的取值范围是{1, -1}，1 意为用户接受推荐，-1 意为用户拒绝推荐；
2) *测试集*
   格式与训练集相同，而（结果）的取值置为 0；
3) *Item*
   格式为：（Item ID）\t（Item 种类）\t（Item 关键词），Item 种类用“a. b. c. d”的格式写成，为分类的层级，关键词用字符串“ID 1; ID 2; ⋯ ID N”来表示；
4) *用户个人信息*
   格式为：（用户 ID）\t（出生年月）\t（性别）\t（推送数），出生年月为用户注册时填写，性别取值{0, 1, 2}，分别意为“未知”、“男”或“女”，推送数为整数，记录了用户推送消息的数量；
5) *用户行为*
   格式为：（用户 ID）\t（用户目标 ID）\t（行为数）\t（分享数）\t（评论数）；
6) *用户 SNS 社交关系*
   格式为：（关注者 ID）\t（被关注者 ID）；
7) *用户关键词*
   格式为：（用户 ID）\t（关键词）。

通过在 7 个数据集中有效的分析、清洗和合并，我们能够得到适合于训练/测试的两个数据集。预期我们将在训练集上计算逻辑回归参数，并将参数代入逻辑回归模型，最终应用在测试集上。预期的结果即与测试集一一对应的一系列推荐 Item ID，这些 Item 即针对用户的个性化推荐结果。

评价方式可采用用户反馈制度，在线收集用户意见，用以评价推荐系统的好坏。

III. 技术方案

在解决问题过程中，我们主要采取了逻辑回归的方法。

逻辑回归类似于多重线性回归，而区别在于它们的因变量不同。两种回归都同属于广义线性模型（Generalized linear model）。逻辑回归的因变量可为二分类或多分类，它的主要应用有：寻找危险因素、预测以及判别。

本文为北京理工大学数据挖掘课大作业项目报告。如有疑问请联系作者，联系方式为：
宁小东，黄建峰，王新灵，图像与可视化团队，中心教学楼 820（e-mail: xdning@bit.edu.cn；hjf579068@qq.com；2213778753@qq.com）。
项目代码可以参见 https://github.com/orangeNya/bitdm_project_final。

在本文中，我们使用逻辑回归进行预测。对于每个样本（记录了用户属性的向量），预测其分类概率，找到概率最大的所属 Item ID，将其作为推荐 Item。

A. 逻辑回归的一般步骤

回归问题的常规步骤为：寻找 h 函数（预测函数）；构造 J 函数（损失函数）；求取使得 J 函数最小情况下的回归参数 θ。

首先构造预测函数 h：预测函数需要借助 Sigmoid 函数，其形式为：

$$g(z)=\frac{1}{1+e^{-z}} \tag{1}$$

构造预测函数为：

$$h_\theta(x)=g(\theta^T x)=\frac{1}{1+e^{-\theta^T x}} \tag{2}$$

其中 θ 为：

$$\theta_0+\theta_1\theta_1+,...,+\theta_n x_n=\sum_{i=1}^{n}\theta_i x_i=\theta^T x \tag{3}$$

那么，对于输入 x 分类结果为类别 1 和类别 0 的概率分别为：

$$P(y=1\mid x;\theta)=h_\theta(x) \tag{4}$$

$$P(y=0\mid x;\theta)=1-h_\theta(x) \tag{5}$$

对于本例多分类问题，可对于所有的备选 item 分类，找出其 $h_\theta(x)$最大的分类，最终逐步获取最适应的 item。

第二步为构造损失函数 J：先使用最大似然估计推导损失函数 Cost，则 J 函数为：

$$\begin{aligned}J(\theta)&=\frac{1}{m}\sum_{i=1}^{n}\mathrm{Cost}(h_\theta(x_i),y_i)\\&=-\frac{1}{m}\left[\sum_{i=1}^{n}y_i\log h_\theta(x_i)+(1-y_i)\log(1-h_\theta(x_i))\right]\end{aligned} \tag{6}$$

最后为寻找回归参数 θ。使用梯度下降法求 θ，梯度下降流公式为：

$$\frac{\delta}{\delta\theta_j}J(\theta)=\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i)-y_i)x_i^j \tag{7}$$

则更新过程为：

$$\theta_j:=\theta_j-\alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x_i)-y_i)x_i^j \tag{8}$$

通常情况下，为了避免过拟合，我们需正则化损失函数，更新模型权重。本项目采用了 Matlab 自带的 optimset 函数优化正则项，并在迭代中反复优化，正规化损失函数 J。

B. 算法设计

基于以上算法原理，我们设计算法 1 来完成预测任务。整个任务分为：构造 J 函数和构造预测函数两大部分。构造 J 函数需构造损失函数并正规化；构造 h 函数即预测任务，需通过 J 函数及 θ 参数得到 sigmoid 函数，同时进行预测。设计的代码架构请参考最终报告文档。

算法 1：逻辑回归

Step1：构造预测函数 h
　　计算边界 $\theta^T h$，后遭函数 $h_\theta(x)$
Step2：构造损失函数 J
　　通过最大似然估计取得对数似然函数 l(θ)
　　则 J(θ) = -1/m * l(θ)，其中-1/m 为系数
Step3：使用梯度下降法求最小值，
　　获取式 $\delta/\delta_\theta j$ J(θ)迭代得出最终的 θ 即结果

IV. 方法实现

方法实现共分为 4 个步骤：数据分析、数据清洗、数据合并以及逻辑回归（数据预测）。

A. 数据分析

对训练集等共 7 个数据集进行数据摘要和可视化分析（见中期报告）后，分析可得其中可用的 18 个数据字段如表 I 所示。

B. 数据清洗

针对训练集、用户行为、用户关键词和用户个人信息 4 个数据集中的无效数据，我们进行了数据清洗。其中包含：

1) 训练集
Item 是指微博中的特定用户，可能是个人、组织或集体，用于推荐给其他用户。例如，名人或知名组织可能会作为备选的“Item 集合”推荐给用户；

2) 用户行为
是指用户将信息上传到微博系统的行为，或者是指推文本身；

3) 用户关键词
用户可以对推送（Tweet）进行评论。评论不会像推送或分享（Retweet）那样显示给他的关注着，而是出现在该推送的评论历史中；

4) 用户个人信息
如果用户 B 关注了用户 A，那么 B 就是 A 的关注者，A 是 B 的被关注者。

C. 数据合并

为了获取适合于训练的数据集，我们对多个数据文件进行了合并。以相同用户 ID、Item ID 或关注者 ID 属性为关系，联立 18 维特征向量（X0 为序号，不具有实际意义；通过合并数据集获得 X1~X17）用于训练和测试过程。提取步骤如图 1 所示。

在合并步骤后，获取用于训练和测试的处理后数据集，并进行下一步预测步骤。

D. 逻辑回归

基于以上算法设计，我们构建功能层级架构如图 2 所示；将其抽象为图 3 的函数架构，并最终实现各函数（见表 II）。

表 I
4.08GB 数据集中可用的数据字段

字段名	数据类型	含义
X0	INT	用于标记数据条数的常数，不参与训练/测试的计算过程
X1	INT	用户推送条数
X2	DOUBLE	用户感兴趣关键词或 item 关键词的加权求和
X3	INT	用户感兴趣关键词或 item 关键词的总数
X4	DOUBLE	用户所艾特的其他用户的加权求和（权重为来往行为次数）
X5	DOUBLE	用户所分享的其他用户的加权求和（权重为来往行为次数）
X6	DOUBLE	用户所评论的其他用户的加权求和（权重为来往行为次数）
X7	INT	用户所艾特的其他用户的总数
X8	INT	用户所分享的其他用户的总数
X9	INT	用户所评论的其他用户的总数
X10	INT	用户所关注其他用户的总数
X11	DOUBLE	用户所艾特的其他用户（包含目标 item 的情况）的加权求和
X12	DOUBLE	用户所分享的其他用户（包含目标 item 的情况）的加权求和
X13	DOUBLE	用户所评论的其他用户（包含目标 item 的情况）的加权求和
X14	INT	用户所艾特的其他用户（包含目标 item 的情况）的总数
X15	INT	用户所分享的其他用户（包含目标 item 的情况）的总数
X16	INT	用户所评论的其他用户（包含目标 item 的情况）的总数
X17	INT	用户所关注其他用户（包含目标 item 的情况）的总数

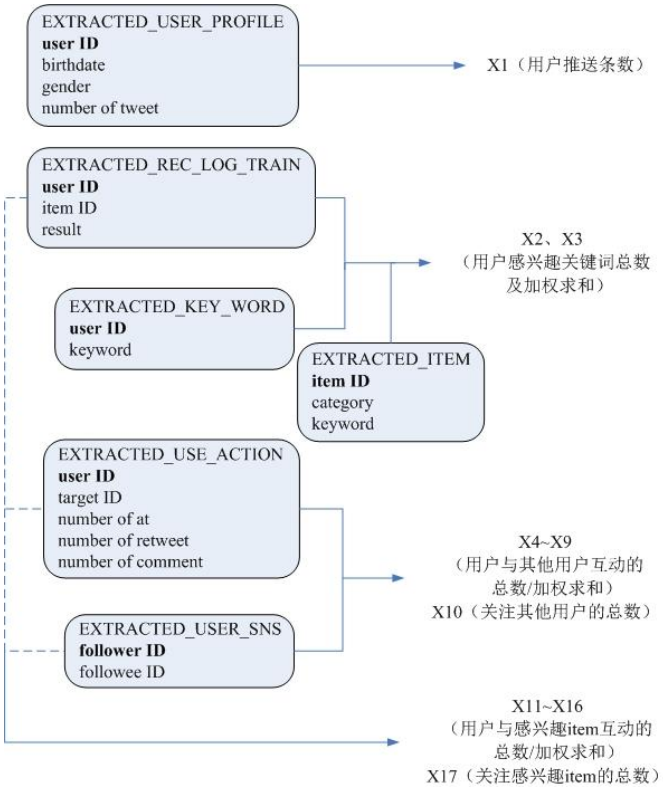

图 1 数据合并的流程图。将数据集按照用户 ID、Item ID 或关注者 ID 进行联立，合并得到 18 维训练/测试数据集。

V. 实验结果

运行主函数可得实验结果，即推荐 Item ID 数据集。实验结果保存在 test_full_y.csv 中，考虑数据分布，我们可得到如下结论：推荐 Item ID 集中在约 200000~1800000 范围内，最频繁的推荐 Item ID 是 ID 为 1025571 的 Item。在测试集推荐中，我们共推荐了 406 个 Item。Item 的接受率可在用户反馈中获得，我们将从在线收集的反馈情况中评价本文推荐方法。

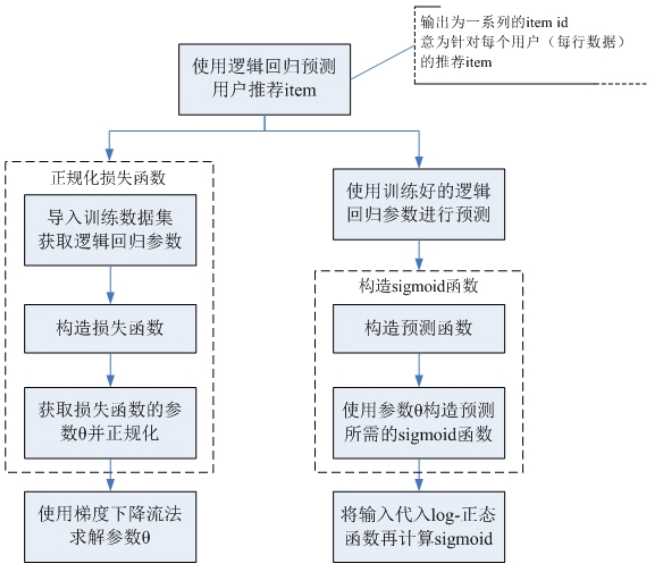

图 2 程序功能层级架构。整个程序分为两个方面：训练和测试。图的左侧分支代表了训练过程，在此过程中我们求得逻辑回归参数，从而得到了逻辑回归模型；图的右侧分支为预测过程，构造预测函数，带入测试数据，从而获取最终结果。

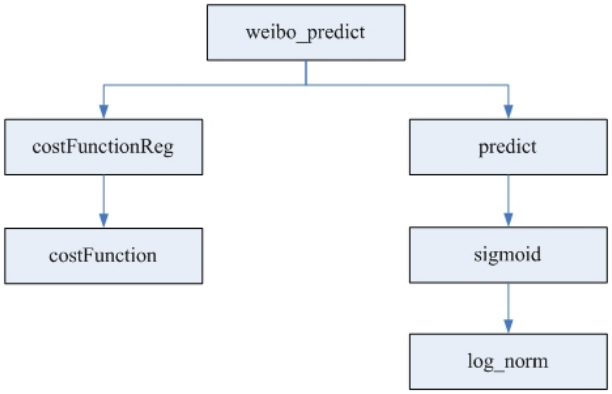

图 3 程序函数层次。将功能层级架构抽象为程序函数层次。左侧和右侧分支分别对应层级架构的相应位置。

表 II
程序各函数功能

函数名称	函数功能
costFunction.m	计算逻辑回归的损失函数
costFunctionReg.m	带正规化的 costFuncion 函数
log_norm.m	log-正态分布函数，用于计算 sigmoid
mapFeature.m	备用的特征映射函数。当增加训练集每条信息的特征维度时，可应用本函数降维
plotData.m	备用的可视化函数。将一组 X 和 y 数据进行可视化，本程序中是将测试集作为 X，预测结果作为 y
plotDecisionBoundary.m	同上，添加了精度确界 theta
predict.m	用于预测的函数，调用了 sigmoid 函数
sigmoid.m	逻辑回归中的 sigmoid 函数
weibo_predict.m	主函数，运行获取最终的推荐 item 列表