数据挖掘大作业最终报告

——选题《社交网络中的个性化推荐系统》

宁小东 2120151024
黄建峰 2120150994
王新灵 2120151042

1. 问题描述及数据集属性

抓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。

数据集大小：4.08GB；

包含文件：

文件名称	含义
item.txt	Item 列表。
rec_log_test.txt	测试用数据集。
rec_log_train.txt	训练用数据集。
user_action.txt	用户行为。
user_key_word.txt	用户关键词。
user_profile.txt	用户个人信息。
user_sns.txt	用户 SNS。

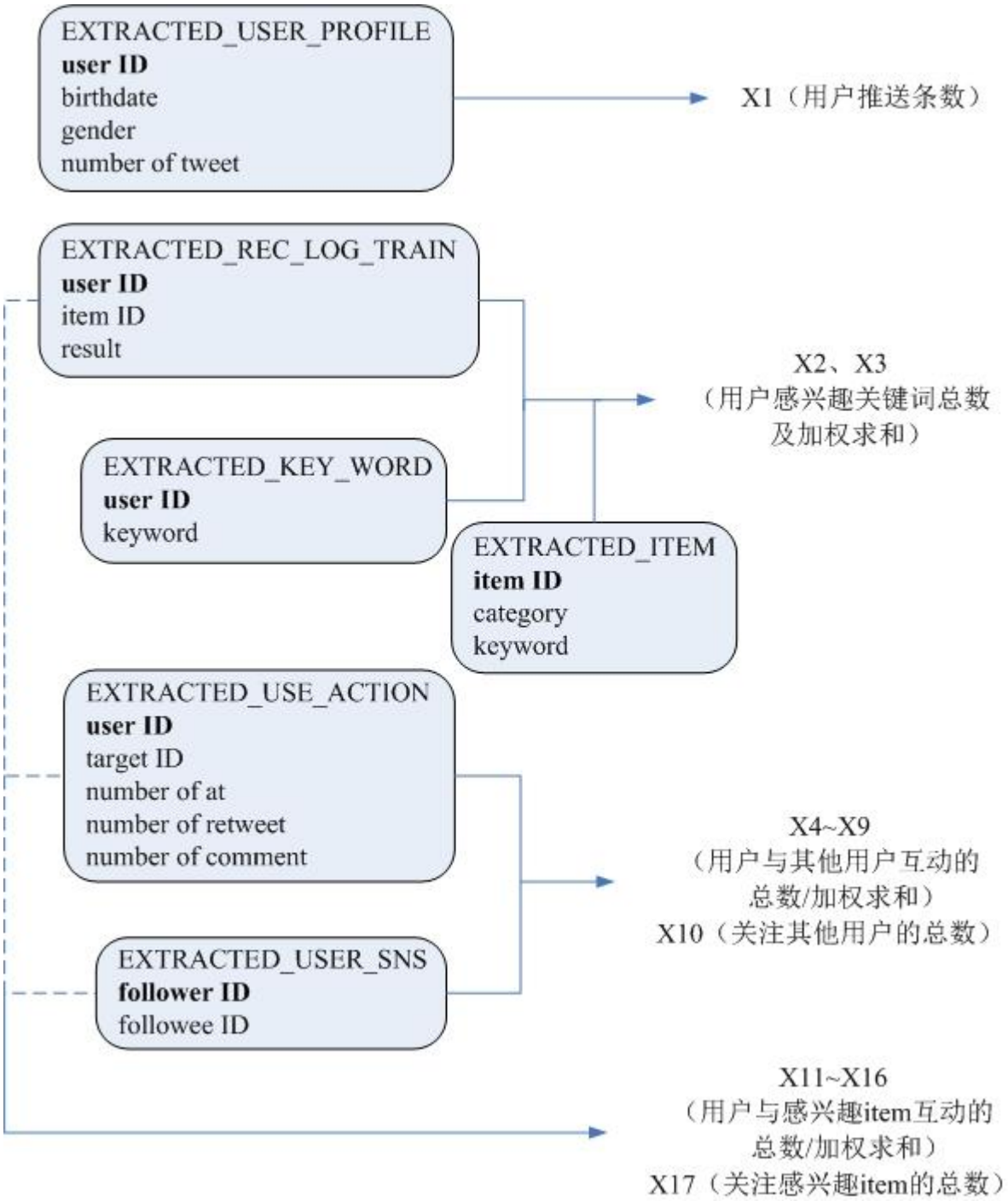
数据清洗后的文件：

文件名称	含义
extracted_rec_log_train.csv	清洗后的训练用数据集，淘汰了部分拒绝比例较高的负样本。
extracted_user_action.csv	清洗后的用户行为，对用户行为存在非法记录的用户进行了清洗。
extracted_user_key_word.csv	清洗后的用户关键词，删除了分类错误的关键词。
extracted_user_profile.csv	清洗后的用户个人信息，删除了出生年月不合法的用户信息。

其他未经处理的文件统一命名为 extracted_X.csv，X 代表对应原 txt 文件名。

2. 数据集合并

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 user id 属性为主键，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。提取步骤如下图所示：



对测试集整合时采取相同的策略，将上述步骤中的 EXTRACED_REC_LOG_TRAIN 变为 EXTRACTED_REC_LOG_TEST。

将数据集整合为 train_log_demo.csv 以及 test_log_demo.csv 以进行进一步处理。

3. 算法原理及设计

Logistic 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）。Logistic 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：寻找某一疾病的危险因素等；

- 预测：根据模型，预测在不同的自变量情况下，发生某病或某种情况的概率有多大；
- 判别：与预测类似，根据模型判断某人属于某病或某种情况的概率有多大。

在本例中，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。

3.1 逻辑回归的一般步骤

Regression 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：

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

构造预测函数为：

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

其中 θ 为：

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

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：

$$P(y=1\mid x;\theta)=h_\theta(x)$$
$$P(y=0\mid x;\theta)=1-h_\theta(x)$$

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：

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

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：

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

则更新过程为：

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

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。

3.2 算法设计

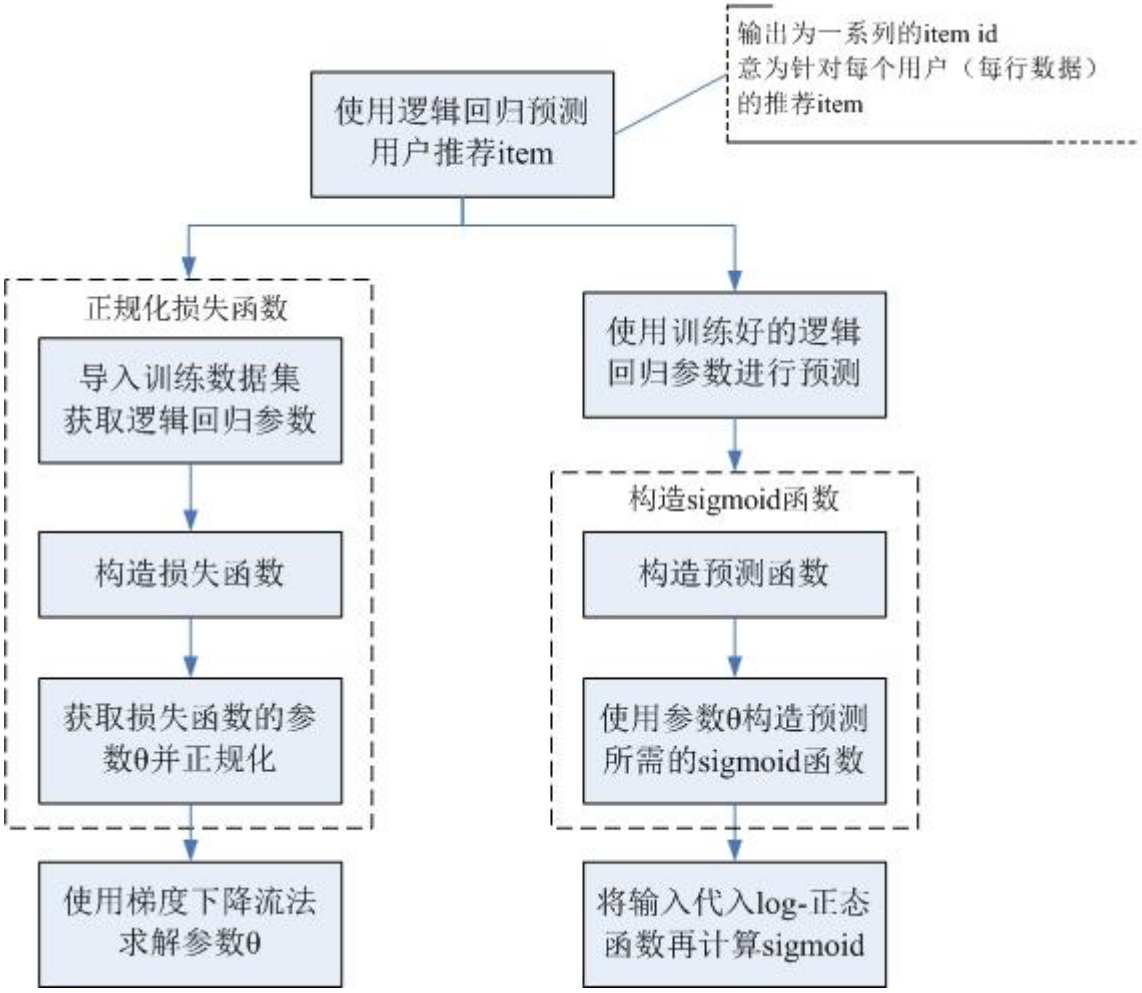
基于以上算法原理，我们设计如下算法来完成预测任务：

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

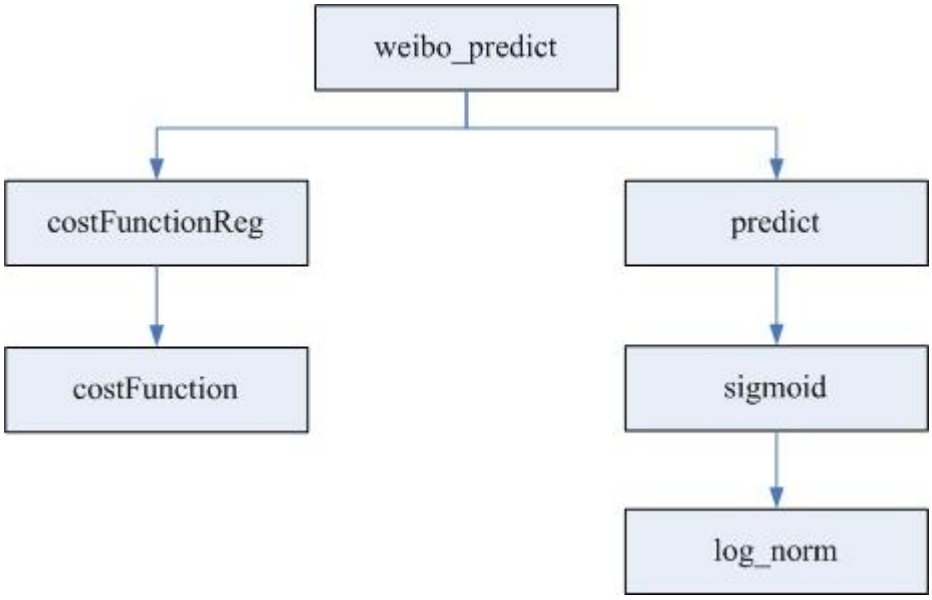
整个任务分为：构造 J 函数和构造预测函数两大部分。构造 J 函数需构造损失函数并正规化；构造 h 函数即预测任务，需通过 J 函数及 θ 参数得到 sigmoid 函数，同时进行预测。

4. 代码架构

基于以上算法设计，本项目的各项功能层次架构如下：



将其抽象为函数调用关系为：



5. 代码说明及实验结果

本项目运行环境为 windows 7 32 位操作系统，处理器为 Intel Core i5-2410M CPU @ 2.30GHz，内存 4.00GB，运行平台为 Matlab R2012a，无其他依赖库或配置文件。

项目共包含 12 个文件，其中 M 函数文件 9 个，.csv 文件（数据集文件）3 个。每个函数的说明如下：

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。

3 个数据集文件分别为：

数据集名称	含义
test_log_demo.csv	合并后的测试数据集，每行数据包含 18 维特征。
train_log_demo.csv	合并后的训练数据集，同上。
test_full_y.csv	预测的推荐 item 列表，每行对应一个 test_log_demo 中的用户。

为了便于代码上传，我们采用随机抽样的方式将 test_log_demo.csv 精简为 30000 条训练数据；test_log_demo.csv 精简为 10000 条。

运行主函数 weibo_predict.m 后，程序自动载入训练数据集 train_log_demo，并对测试

机 test_log_demo 进行回归分析，获得最终结果 test_full_y。

整个计算过程显示如下：

```
Regularization finished.

ans =

  Columns 1 through 8

   -2.7851    0.0422    0.0754   -0.0918   -0.3359    0.0948   -0.7584    0.2179

  Columns 9 through 16

   -0.5085    0.2052    0.1878   -0.3947    0.1359    0.6784   -0.2441    0.5609

  Columns 17 through 18

    0.5832    0.0570

Loading test file...
Test file loaded.
Logrocessing...
Computung prediction...
Testing data prepared.
Recall: 9  0.09%
Train Accuracy: 99.910000

Start saving result...
All finished.
```

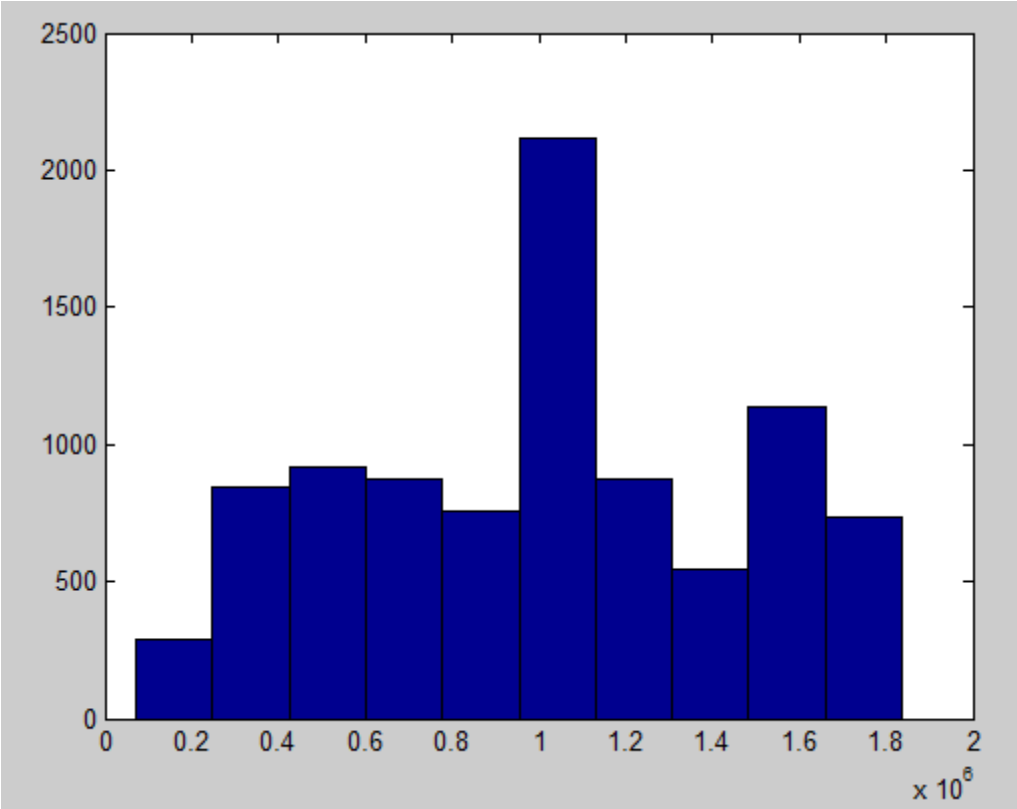
上图显示了各列正规化系数以及推测的准确率。

部分实验结果如下图所示（为推荐 item 的 id 列表）：

```
207545
208251
208392
208692
208710
208745
209574
209786
210068
210174
210579
211020
211461
211691
211744
```

6. 结论

在本微博 item 推荐问题中，我们首先通过清洗、合并用户信息等数据集，获取了用于预测的特征向量集合，并对每一组向量（对应一个用户）进行了预测，最终得到了每个用户的推荐 item。对结果进行分析，可得其推荐 item 分布：



其中推荐 id 众数为：

```
Most recommand frequency item id: 1025571
```

能够看出测试样本中推荐 item id 集中在约 200000~1800000 范围内，最频繁的推荐 item 是 id 为 1025571 的 item。在本次推荐中，共包含：

```
406 items in total.
```

406 个 item。