華東理工大學

模式识别大作业

题　　目　Criteo 展示广告

学　　院　信息科学与工程

专　　业　控制科学与工程

组　　员　方华

指导教师　赵海涛

完成日期：　2019 年 11 月 28 日

Criteo 展示广告：预测用户是否会点击广告

组员：方华

经过前十周的理论学习，《模式识别原理与应用》课程已落下帷幕，但作为一门实践与理论相结合的课程，对模式识别的理解和运用不应只停留在理论学习方面，更应通过实践来获得提高，如此，不仅可以加深对理论认识的理解，而且锻炼了自身解决问题能力。

本次作业来自 lintcode，通过题目已给出的数据，建立更好的点击率预测模型，从而预测用户在不同的特征下是否会点击广告。根据题目要求，本题需要用到随机森林（Random Forest）和逻辑回归（Logic Regression）的相关知识，而这部分恰好是刚刚所学的内容，因此，选择该题不会超出自身目前所掌握的理论知识，并且能够对已学的知识有较好地巩固。

一、问题描述

Criteo 是一家第三方的展示广告公司，与世界上超过 4000 家电子商务公司有合作关系。说到广告，关注的最多的就是点击率了。生活中，经常能听说某人通过建立更好的点击率预测模型，为公司带来上亿的增量收入。

本题使用 Criteo 所共享的一周展示广告数据，数据中提炼了 13 个连续特征、26 个离散特征和用户是否点击了该页面广告的标签。本文所要解决的问题就是如何训练出合适的模型，预测用户在不同的特征下是否会点击广告。

二、整体解决方案

2. 1 问题分析

问题所给出的训练数据文件 train.csv 提供了 1599 条的用户访问网页和点击广告记录的对应特征，其中：I1～I13 为计数特征，C1～C26 为类别特征，Label 表示用户是否点击广告，0 为未点击，1 为点击。因此，对于一个二分类问题，并且在训练数据中是含有标签的，可以利用有监督学习从标签化训练数据集中推断出函数的学习任务，再用相应的算法建立模型来预测未知样本。

有监督学习任务主要分为以下几个步骤：1）数据预处理；2）利用学习算法训练模型；3）验证模型；4）预测新数据。有监督学习任务的模型流程图如图 1 所示。

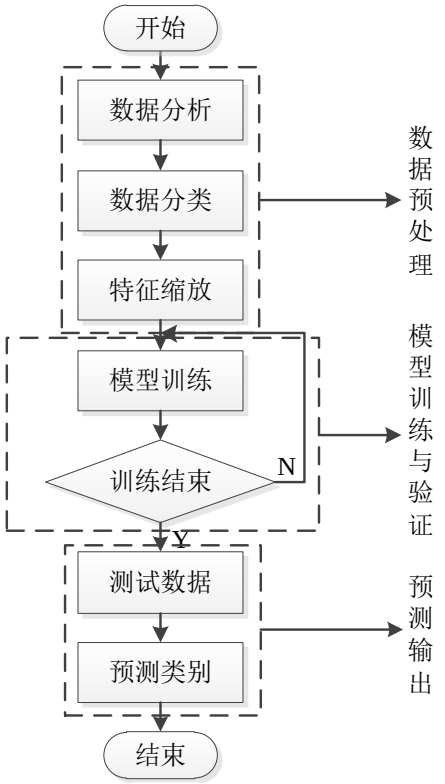


图 1 有监督学习任务的模型流程图

2. 2 数据预处理

2. 2. 1 数据分析

图 2 和图 3 分别为训练数据文件（train.csv）的计数和类别数据（部分）。

Id	Label	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12	I13
10000842	1		48	13	12	2940	775	2	7	193		0		12
10000962	0	2	-1			285	5	2	1	22	2	2		
10000481	0		241	13	12	502		0	16	88		0		12
10000268	0		2242	13	7	11786	63	29	13	87		1		13
10000711	1		30	1	1	30227	8	3	6	9		1		1
10000031	0		-1			4956		0	37	97		0		
10000883	0		30	3	18	11776		0	24	129		0		18
10001971	0	6	17	2	12	11	12	6	2	12	1	1		12
10001349	0	0	94	3	3	5955		0	3	3	0	0		3
10000478	0	20	-1	1	2	2	2	20	2	2	1	1		2
10001307	0	0	392		1	1463	20	43	1	8	0	3		1
10001425	0		50	12	13	10506	132	5	8	152		4		13
10001411	0	1	584		12	135	12	1	13	312	1	1		12
10000746	0	0	754	16	0	4845	507	29	2	115	0	9	0	1
10001878	0	9	103	18	43	79	54	9	48	50	1	1		48
10000347	1	6	0	23	0	5	0	11	0	0	1	2		0
10001487	0	3	271		0	4	1	3	0	0	1	1		0
10001772	0		-1						0					
10001273	0	0	322	16	33	3218	282	4	27	337	0	2		34
10001543	0		2		1	33213	111	9	1	38		0		1
10000937	0		-1			1172		0	46	45		0		
10000935	0		-1						0					
10000900	1		1		5	42642	164	0	5	153		0		5
10001103	0	1	0	45	15	15	10	5	13	240	1	4		10
10000489	1	0	0	15	10	1491	151	10	31	205	0	4	7	10
10000089	0		1	75	22	5912	181	2	22	68		1		22
10001837	0	0	3	1	6	30860		0	19	15	0	0		6
10000768	0	2	1	36	4	384	42	6	7	65	1	4		4
10001458	1		25	63	26	20265	144	3	13	36		2		26
10001460	0		9			7414	172	1	0	19		1		
10000177	0		1	5	4	2931	36	2	6	62		1		5
10000311	0	0	225	5		1444	32	7	3	12	0	4		

图 2 训练计数数据（部分）

Id	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25	C26
10000743	51b97b8f	b28479f6	d345b1a0	3fa658c5	3486227d	e88ffc9d	c393dc22	b1252a9d	57c90cd9		bcdee96c	4d19a3eb	cb079c2d	456c12a0
10000159	ab8a1a53	07d13a8f	06969a20	9bc7fff5	07c540c4	9.3E+07			242bb710		3a171ecb	72c78f11		
10001166	18fc2b1e	cfef1c29	dad721df	1f2e9dec	07c540c4	25c88e42	21ddcdc9	b1252a9d	a0136dd2		32c7478e	8fc66e78	001f3601	f37f3967
10000318	ee79db7b	07d13a8f	36721ddc	fc60350c	e5ba7672	5aed7436	21ddcdc9	b1252a9d	c3abeb21		423fab69	1793a828	e8b83407	5cef228f
10000924	b0bfed6d	07d13a8f	3b2d8705	74d50e5e	e5ba7672	642f2610	1d1eb838	b1252a9d	1640d50b	ad3062eb	423fab69	45ab94c8	2bf691b1	c84c4aec
10000532	6726c9f7	b28479f6	e1ac77f7	b041b04a	e5ba7672	2804effd			723b4dfd		32c7478e	b34f3128		
10000485	5978055e	########	19f03519	636ac9fa	8efede7f	1e3d9f94			a5f11cda		3a171ecb	732eaaee		
10000251	654099a3	07d13a8f	6afa614f	b9660ab7	3486227d	a4667218			77e124f1	c9d4222a	55dd3565	38be899f		
10001288	e6fc496d	1adce6ef	c06bba41	ddfabc04	3486227d	15a4e6dd	21ddcdc9	5840adea	87e212d4		32c7478e	c2fe6ca4	001f3601	602f0609
10001697	b0c30eeb	07d13a8f	22223d6c	d388d33c	d4bb7bd8	97b81540			0ac4575d		be7c41b4	d28d80ac		
10000190	879fa878	b28479f6	a66dcf27	31ca40b6	d4bb7bd8	7b49e3d2			dfcfc3fa		423fab69	aee52b6f		
10001836	5781932	1adce6ef	a5007c7b	6de617d3	07c540c4	4771e483			df66957b		3a171ecb	b34f3128		
10000615	e97e78ef	07d13a8f	760e4845	02af86f2	e5ba7672	1c437c51			7812b499		32c7478e	0cae79aa		
10001665	8E+07	07d13a8f	2d5d7e11	4f264080	2005abd1	98c4d3e0			91e1efa5		55dd3565	7fb4ff91		
10001500	eb11180c	b28479f6	2530f036	48e224f3	07c540c4	3e608631			a9fb024b	8ec974f4	32c7478e	6e9a987f		
10000351	8E+07	b28479f6	ad31dfdb	2a7f62df	2005abd1	082ff924			b691bac2		bcdee96c	88ed380d		
10001217	18a5133f	b28479f6	f511c49f	1203a270	07c540c4	752d8b8a			73d06dde		bcdee96c	aee52b6f		
10001524	7301027a	07d13a8f	3b2d8705	2235af84	3486227d	642f2610	1d1eb838	a458ea53	5df9af5f		423fab69	45ab94c8	2bf691b1	c84c4aec
10001361	1211c647	b28479f6	ab07a5b8	0c6b4ad6	e5ba7672	d5d1ae1c			f6e3bd9c		32c7478e	9e07eb4a		
10000740	85dbe138	b28479f6	ac182643	7a76439d	8efede7f	1f868fdd	ea132b7c	a458ea53	d7be7b8d		32c7478e	9b786035	9d93af03	72ecd2e2
10001087	009f5dd8	1adce6ef	8ff4b403	01adbab4	3486227d	26b3c7a7			21c9516a		423fab69	b34f3128		
10000253	d5aaf8c3	1adce6ef	dbc5e126	28fa6301	e5ba7672	5aed7436	3a52a22f	a458ea53	059d6830		423fab69	54761a25	ea9a246c	8e3c920c
10001432	c679a49f	cfef1c29	131dff63	2e97807d	07c540c4	cc693e93	21ddcdc9	5840adea	9ce3d12f	c9d4222a	32c7478e	03ba7282	ea9a246c	983e320a
10000299	9259d03d	b28479f6	a785131a	aafa191e	27c07bd6	005c6740	21ddcdc9	5840adea	7e5b7cc4		3a171ecb	1793a828	e8b83407	b9809574
10001920	e1b62f8f	07d13a8f	41f10449	bad5ee18	07c540c4	698d1c68			0429f84b	c9d4222a	be7c41b4	c0d61a5c		
10000568	5e419718	b28479f6	03b0a8e3	ebc92e5f	07c540c4	4ac4fd60			9e805f53	ad3062eb	85d5a995	01eb9c81		
10000838	f3747bf1	1adce6ef	d9e19f11	01adbab4	e5ba7672	e32bf683			21c9516a		32c7478e	b34f3128		
10001287	e9332a03	07d13a8f	5cedaf14	d14a4197	e5ba7672	c04ce6df	6f86de17	a458ea53	e3b993b8		3a171ecb	9117a34a	001f3601	54ca28ff

图 3 训练类别数据（部分）

从图中可以很明确的看出，无论是计数特征还是类别特征都存在着缺失值，因此在选择训练数据的过程中需要进行取舍。

训练数据特征的选取本文采用了两种思路，第一种是仅选择前十三组的数值数据作为训练样本，因为前十三组的特征都为数字量，不含有字母，较为方便进行训练和处理；第二种思路是加入部分类别量进行训练，但是会涉及到非数字量的特征，因此需要将这些信息转化为 0/1 的量。

同样地，测试数据（test.csv）也存在着缺失的问题，所以需要对训练集和测试集都进行数据的分类，而由于各个特征所处的值域范围不同，为了确保这些不同的特征能够处在一个相近的范围内，考虑对各个特征进行特征缩放。

2.2.2 数据分类与预测

数据分类主要运用随机森林法，随机森林在分类问题中有广泛地应用：在数据集上表现良好；在当前的很多数据集上，相对其他算法有着很大的优势；它能够处理很高维度（特征）的数据，并且不用做特征选择，下面简单介绍其原理。

随机森林指的是利用多棵树对样本进行训练并预测的一种分类器。随机森林分类（RFC）是由多种决策树分类模型 $\{h(X,\theta_k),k=1,2,\cdots,N\}$ 组合而成的一种模型，且参数 θ_k 是独立同分布的随机变量。在给定自变量集合 X 下，每个决策树分类模型都有投票权来选择最优的分类结果。随机森林分类的基本过程包括：利用自助法（bootstrap）从原始训练集抽取 N 个样本，且每个样本的样本容量与原始训练集一样；对 N 个样本分别建立 N 个决策树模型，得到 N 种分类结果；根据 k 种分类结果对每个记录进行投票表决决定其最终分类。

随机森林通过构造不同的训练集增加分类模型间的差异，从而提高组合分类

模型的预测能力。通过 K 轮训练，得到一个分类模型序列 $\{h(X,\theta_k),k=1,2,\cdots,N\}$，再用它们构成一个多分类模型系统，该系统的最终分类结果采用简单多数投票法。最终的分类决策：

$$H(x)=\arg\max_{r}\sum_{i}I\left(h_i(x)=Y\right)$$

其中，$H(x)$ 表示组合分类模型，$h_i(x)$ 是单个决策树分类模型，Y 表示输出变量（或称目标变量），$I(*)$ 为示性函数。该公式说明了使用多数投票决策的方式来确定最终的分类。

因此，建立相互关联性小且强度高的随机森林，要解决的问题有：1）如何构建随机决策树；2）如何对每棵树进行有效地组合。

首先是随机样本的选取。随机森林是决策树的组合，先用装袋算法（bagging）产生不同的训练集，也就是从原始训练集中利用自助法（bootstrap）抽样生成新的训练集，对每个新的训练集利用随机特征选取方法生成决策树，且决策树在生长过程中不进行剪枝。通过观察，原始训练集中接近 37%的样本出现在自助法所采集的样本中，这些数据为袋外数据（OOB），使用这些数据来估计模型的性能称为 OBB 估计。因此，随机森林本质上是装袋算法的一个扩展变体。

其次是随机特征选取。为了使得每棵树之间的关联尽可能地小，在构造树时要对它的特征进行恰当的选择，主要有两种方法：随机选取特征变量和随机选取特征变量的线性组合。

最后随机决策树的构建：在构建随机决策树时，对采样后的数据使用完全分裂的方式建立决策树，这样，决策树的某一个叶子节点要么是无法继续分裂的，要么里面的所有样本都是指向同一个分类。具体构造过程为：

1）确定一个区别度最大的属性；

2）根据所确定的属性创建一个树节点和相应的子节点，且每个子节点所表示的是所选属性的一个唯一的取值；

3）重复步骤 1、2，直到这样决策树的某一个叶子节点要么是无法继续分裂，要么里面的所有样本都指向同一个分类。

综上，便是随机森林法的原理。

针对本题所给出的训练数据，通过观察可以发现，I2 列的数据是齐全的，I5,I6,I7,I8,I9,I11 这 6 列数据缺失的较少，为了保证随机森林算法的实施，对这 6 列特征进行平均值法进行补全，而剩下的 I1,I3,I4,I0,I12,I13 这几列特征采用随机森林的方法进行特征值预测，同时每完成一次随机森林算法，下一次随机森林算法的数据都会增加一列，从而呈现出一个逐渐完善数据的过程。

随机森林法流程图如图 4 所示。

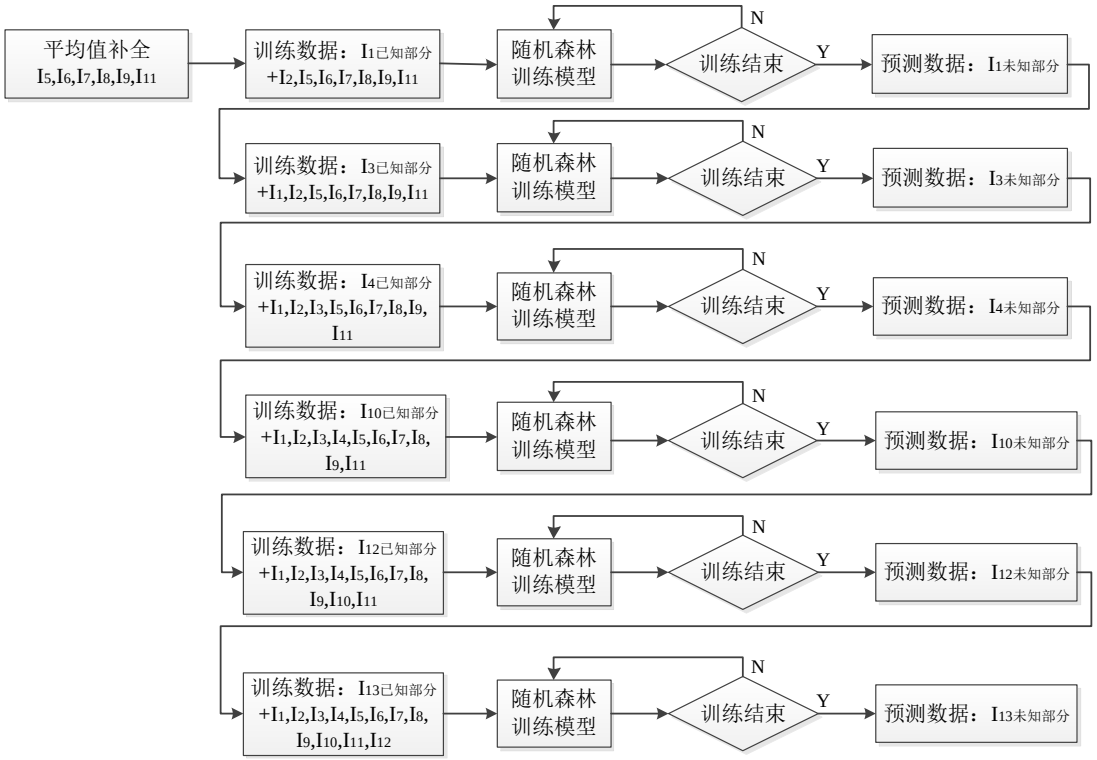


图 4 随机森林法流程图

程序代码如下：

```
def set_missing_I1s(df): # 把已有的数值型特征取出来放进 Random Forest Regressor 中
    I1_df = df[['I1','I2','I5','I6','I7','I8','I9','I11']]
    known_I1 = I1_df[I1_df.I1.notnull()].as_matrix()
    unknown_I1 = I1_df[I1_df.I1.isnull()].as_matrix()
    # y 即目标
    y = known_I1[:, 0]
    # X 即特征属性值
    X = known_I1[:, 1:]
    # fit 到 RandomForestRegressor 之中
    rfr = RandomForestRegressor(random_state=0, n_estimators=2000, n_jobs=-1)
    rfr.fit(X, y)
    # 用得到的模型进行未知结果预测
    predictedI1s = rfr.predict(unknown_I1[:, 1::])
    # 用得到的预测结果填补原缺失数据
    df.loc[ (df.I1.isnull()), 'I1' ] = predictedI1s
    return df, rfr
```

```
X_train ,rfr = set_missing_I1s(X_train)
tmp_df = X_test[['I1','I2','I5','I6','I7','I8','I9','I11']]
null_I1 = tmp_df[X_test.I1.isnull()].as_matrix() #  根据特征属性 X 预测并补上
X = null_I1[:, 1:]
predictedI1s = rfr.predict(X)
X_test.loc[ (X_test.I1.isnull()), 'I1' ] = predictedI1s
```

2.2.3 特征缩放

在数据的分类和预测的基础上，再对训练数据进行缩放，以使得这些不同的特征能够处在一个相近的范围内。程序代码如下：

```
[m,n]=size(P);
Pmean=mean(P);%均值
Pstd=std(P);%标准差
for i=1:m
    P1(i,:)=(P(i,:)-Pmean)/(Pstd);%特征缩放
end
```

至此，经过数据预处理后的训练数据如图 5 所示（部分）。

Label	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12	I13
0	-0.28205	-0.2336	-0.11813	0.086562	-0.24391	-0.35441	-0.18314	1.316916	-0.22312	-0.59573	-0.0463	-0.26274	-0.10084
1	-0.28205	-0.27436	-0.09804	-0.04312	-0.24858	-0.45352	-0.12475	-0.20907	-0.45003	-0.59573	-0.33991	-0.07692	-0.03638
0	-0.14629	-0.26983	-0.00157	0.086562	-0.2707	-0.39479	-0.24154	-0.25267	-0.47944	1.202935	-0.33991	-0.0371	-0.01554
1	-0.14629	-0.25398	0.038626	1.538654	-0.27711	-0.50124	-0.24154	-0.07827	-0.34498	1.202935	-0.33991	0.008182	-0.05819
0	-0.19801	-0.27209	-0.10608	2.767347	-0.09296	0.772557	0.595475	1.186118	3.21829	-0.57414	0.540925	-0.22392	0.922686
0	8.270623	-0.26983	-0.03774	-0.47193	-0.27635	-0.52327	0.965319	-0.51427	-0.50886	1.202935	-0.33991	0.111044	-0.31407
0	-0.28205	-0.19284	0.211464	-0.58363	0.521341	0.757873	-0.26101	-0.38347	-0.33658	-0.59573	-0.63352	-0.04445	-0.22878
0	-0.28205	-0.20642	-0.11411	-0.36024	-0.10918	-0.002	-0.24154	-0.20907	-0.36179	-0.59573	-0.33991	-0.26756	-0.27142
0	-0.28205	-0.19057	-0.13823	-0.47193	-0.13029	-0.18922	-0.22208	-0.47067	0.751733	-0.59573	-0.33991	-0.16457	-0.31407
0	-0.28205	-0.20869	-0.13823	-0.24854	-0.22018	-0.4939	-0.24154	-0.33987	-0.49205	-0.59573	-0.33991	-0.08606	-0.22878
1	-0.28205	-0.10226	-0.12617	0.086562	-0.27262	-0.35808	0.556544	0.750121	2.075354	-0.59573	2.302598	-0.29077	0.368277
0	-0.14629	-0.27209	-0.14627	-0.69533	-0.27198	-0.50491	-0.10528	-0.42707	-0.29876	1.202935	0.540925	-0.03672	-0.39936
0	-0.28205	-0.26756	-0.07392	-0.69533	-0.27343	-0.27365	-0.18314	-0.29627	-0.50045	-0.59573	-0.33991	-0.29077	-0.39936
0	-0.20181	-0.27209	0.00647	0.868458	-0.192	-0.24428	0.478682	0.052527	1.600531	-0.57414	0.834537	-0.24561	0.240337
0	0.260981	-0.27209	0.207444	-0.36024	-0.26785	-0.51593	-0.18314	-0.29627	-0.50466	1.202935	-0.33991	0.243586	-0.27142
0	-0.28205	-0.27209	-0.10608	-0.58363	-0.19479	-0.46086	-0.24154	-0.51427	-0.50045	-0.59573	-0.33991	-0.14567	-0.35672
0	-0.14629	-0.04565	-0.14225	-0.58363	-0.27219	-0.3801	0.361889	1.360516	3.525034	1.202935	1.421761	-0.22545	0.496218
0	-0.28205	-0.17019	-0.14627	0.868458	-0.24581	-0.08643	-0.10528	0.488523	0.436585	-0.59573	-0.0463	-0.08808	1.306507
1	1.482791	-0.22907	-0.07794	1.203556	-0.2505	-0.32504	1.00425	1.142518	0.377758	-0.59573	1.421761	-0.0409	0.325631
0	-0.28205	-0.27436	-0.06588	2.543948	-0.04785	0.519265	-0.24154	0.008927	0.676098	-0.59573	-0.33991	-0.25918	0.880039
0	-0.01053	-0.07283	-0.1188	-0.47193	-0.27647	-0.52327	0.206165	-0.60147	0.121437	1.202935	0.834537	-0.05777	-0.31407
0	-0.28205	-0.27436	-0.14627	-0.36102	-0.21526	-0.24795	-0.24154	-0.33987	-0.23573	-0.59573	-0.33991	-0.29077	-0.23718
0	-0.28164	6.677275	-0.14225	-0.45987	0.089304	-0.5196	0.050441	-0.60147	-0.50466	-0.57594	0.247313	-0.15899	-0.26049
0	0.396738	-0.15661	-0.10201	-0.24854	-0.26841	-0.51593	-0.16368	-0.29627	-0.47524	1.202935	-0.33991	-0.0371	-0.22878
0	0.543016	0.090208	-0.09866	-0.13684	0.213792	0.034708	0.731734	-0.33987	0.457595	-0.57414	0.247313	-0.19374	-0.18613

图 5 预处理后的训练数据（部分）

2.3 模型构建

根据题目所给的提示，可以采用逻辑回归（Logistic Regression）算法构建模型。Logistic Regression 的起源主要分为几个阶段，从开始想到 logistic 这个词，到发现 logistic function，再推导出 logistic function，最后才命名 Logistic Regression。

logistic 起源于对人口数量增长情况的研究，逻辑回归作为 Regression Analysis 的一个分支，它实际上还受到很多 Regression Analysis 相关技术的启发。逻辑回归是为了解决分类问题，根据一些已知的训练集训练好模型，再对新的数据进行预测属于哪个类。

2.3.1 原理

类比于线性回归（基于吴恩达老师所讲述的机器学习内容），本文希望逻辑回归模型也有一个假设函数 $H_\theta(x)$，但是对于这个假设函数，本文希望它的取值范围为 $0 \le H_\theta(x) \le 1$，因此，$H_\theta(x)$ 的形式如下：

$$\begin{cases} H_\theta(x) = g\left(\theta^T X\right) \\ g(z) = \dfrac{1}{1+e^{-z}} \end{cases} \Rightarrow H_\theta(x) = \frac{1}{1+e^{-\theta^T X}}$$

通过 matlab 画出 $g(z)$ 的图像如图 6 所示，$g(z)$ 的取值范围为 $0 \le g(z) \le 1$，符合要求。因此，对于 $H_\theta(x)$ 的输出值，本文将它类似的看成一种概率的估计，即输入为 X，并且类别标签 $y=1$ 时的概率。

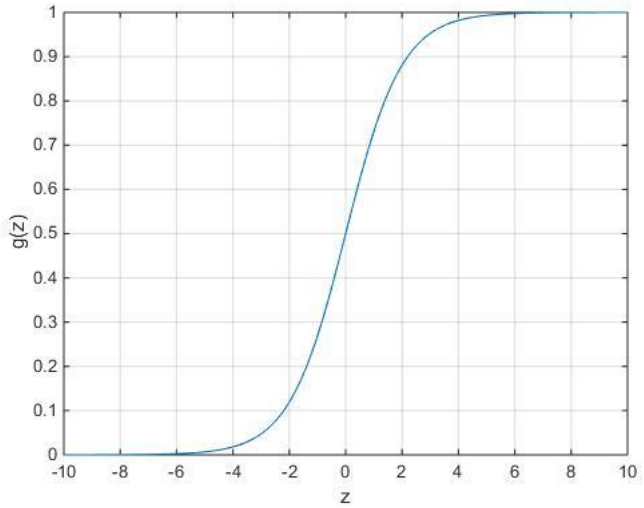


图 6 $g(z)$ 图像

对于二分类问题，只有两个类别，我们选择标签为 0 和 1，0 代表消极类别，1 代表积极类别。例如：$H_\theta(x) = 0.7$ 则表示在输入特征 X 的情况下，属于积极类别的概率为 70%。需要注意的是：特征大多数是多维矩阵，所以此处采用大写 X，在公式中为了和 $H_\theta(x)$ 统一，采用小写 x。因此，采用确切的概率形式表示：

$$H_\theta(x) = P\left(y=1 \middle| x,\theta\right)$$

其含义为：给定输入特征 X，在参数 θ 的作用下，属于类别 $y=1$ 的概率。

$$P\left(y=0 \middle| x,\theta\right) + P\left(y=1 \middle| x,\theta\right) = 1$$

$$P\left(y=0|x,\theta\right)=1-P\left(y=1|x,\theta\right)$$

对于分类问题，最重要的是构建出决策边界。对于$H_\theta\left(x\right)$，由图 6 可知，当$g\left(z\right)\geq 0.5$时，$z\geq 0$；当$g\left(z\right)<0.5$时，$z<0$。因此，可以很明显的看出$g\left(z\right)=0.5$是一个临界值，类似一个分界面，将这个映射到 H 函数上，因此构建如下策略：

$$\begin{cases}H_\theta\left(x\right)\geq 0.5,\theta^T X\geq 0\Rightarrow y=1\\ H_\theta\left(x\right)<0.5,\theta^T X<0\Rightarrow y=0\end{cases}$$

从概率的角度来说，假设当前的目标属于某一类别，因此，当概率大于 0.5 的情况下，判定属于积极类别，当概率小于 0.5 的情况下，判定为消极类别。理论上可行，但需要注意的是，现在所构建的决策边界本质上是假设函数$H_\theta\left(x\right)$的属性，决定的是分类面，与参数θ有关。因此，参数θ如何来获得，将是逻辑回归模型中最重要的一个环节。

想要确定参数θ的值，需要构建一个代价函数，优化的目标就是让代价函数最小，即：在做决策的时候，希望决策者做的决策所付出的代价要尽可能小。对线性回归来说，预测值和真实值之间的误差越小，决策者付出的代价也就越小，拟合的效果也就越好。因此对于线性回归来说，其代价函数如下：

$$J\left(\theta\right)=-\frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(H_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2=-\frac{1}{m}\sum_{i=1}^{m}loss\left(H_\theta\left(x^{(i)}\right),y^{(i)}\right)$$

其中，m为训练集的样本个数。但由于$H_\theta\left(x\right)$是一个非线性函数，若采用上述线性回归的$loss\left(H,y\right)$，则会生成非凸形式的$J\left(\theta\right)$，无法利用梯度下降法收敛到全局最优，因此，需要重新构建一个$loss\left(H,y\right)$如下所示：

$$loss\left(H_\theta\left(x\right),y\right)=\begin{cases}-\log\left(H_\theta\left(x\right)\right) & ,y=1\\ -\log\left(1-H_\theta\left(x\right)\right),y=0\end{cases}$$

图 7 的所描绘的分别是$y=1$和$y=0$的情况下$loss\left(H,y\right)$的图形。其中，横坐标H代表的就是$H_\theta\left(x\right)$，纵坐标$loss$代表的$loss\left(H,y\right)$。

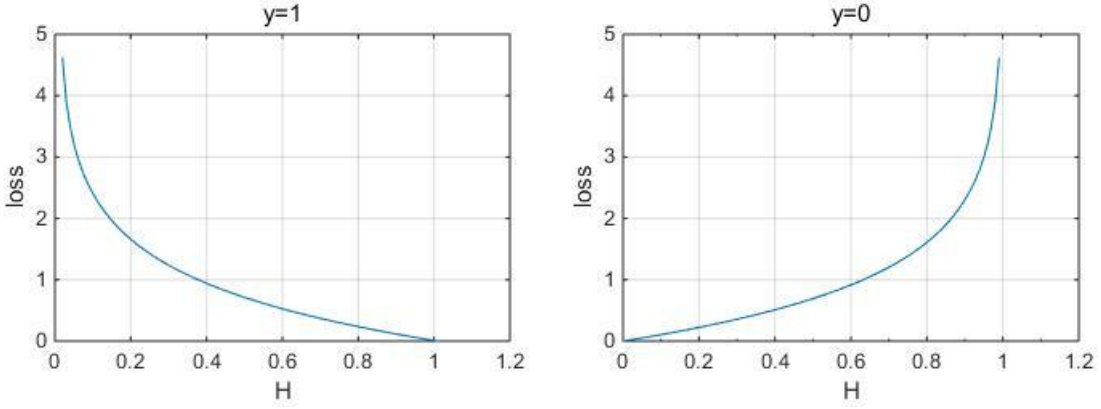


图 7 *loss* 函数在不同 *y* 时*的*图像

从上图可以看出，当 $y=1, H_\theta(x)=1$ 时，$loss=0$，即代价函数也为 0；当 $y=1$，$H_\theta(x)\to 0$ 时，$loss$ 迅速增大，并逐渐趋向于无穷。同理，当 $y=0, H_\theta(x)=0$ 时，$loss=0$；当 $y=0, H_\theta(x)\to 1$ 时，$loss$ 迅速增大，并逐渐趋向于无穷。说明，错误决策的损失非常严重，所以，应尽量避免错误决策。

通过上述分析，所构建的代价函数可以作为目标函数，并将代价函数的最小化作为优化目标，就可以求得较为理想的模型。为了方便后续的优化处理，将上面的两式合并：

$$loss\left(H_\theta(x), y\right) = -y\log\left(H_\theta(x)\right) - (1-y)\log\left(1-H_\theta(x)\right)$$

不难发现，上式即为极大似然估计所得出的一个公式。因为，y 的取值只有两种可能，非 0 即 1，所以，当 $y=1$ 时，第二项中的 $1-y=0$，就消去了第二项；同理，当 $y=1$ 时，也是如此。

同时，可以采用梯度下降法最小话代价函数，从而达到全局最优，即：

$$\min_\theta J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} -y\log\left(H_\theta(x)\right) - (1-y)\log\left(1-H_\theta(x)\right)\right]$$

$$\therefore \theta_j := \theta_j - \alpha\sum_{i=1}^{m}\left(H\left(x^{(i)}\right) - y^{(i)}\right)x_j^i$$

在求出参数 θ 后，将参数带回到最初的，此时就完成了模型的训练，可以使用该模型对测试集数据进行预测。

2.3.2 程序

上述的画图及模型程序均使用 matlab 编程实现，因为画图的代码比较简单，故不在单独列出，下面给出 logistic 回归模型的代码。

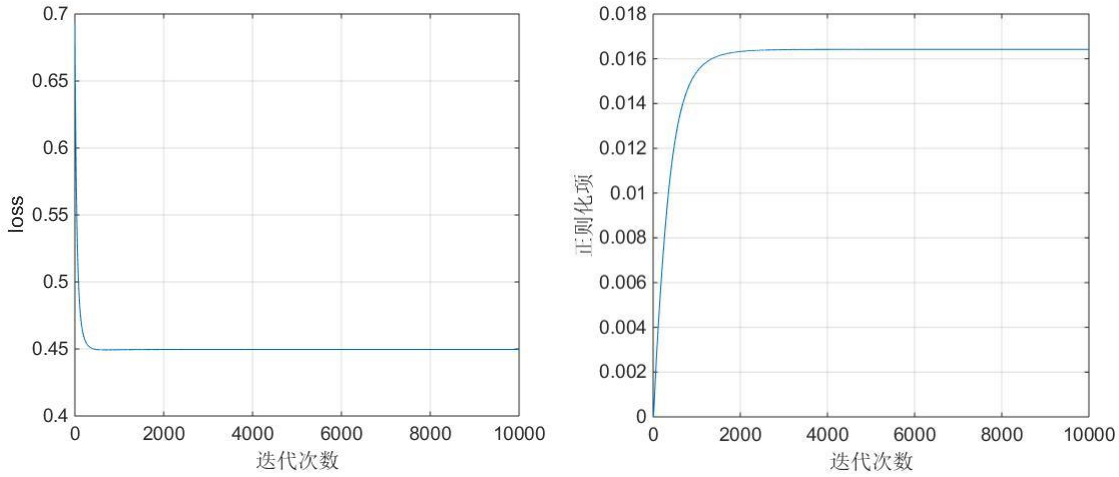
```
%广告点击预测
A=csvread('test4.csv');
label=csvread('label.csv');
[m,dim]=size(A);%特征维度


for i=1:m
A(i,dim+1)=1;
end


X=A(:,1:dim+1);%训练集数据
Y=label;%训练集 label
B=zeros(dim+1,1);%初始化参数矩阵
```

```
step=0;%迭代步数

Z=X*B;
for j=1:m
        H(j,:)=1/(1+exp(-Z(j,:)));%sigmiod 函数
end
E(1,:)=(-1/m)*(Y'*log(H)+(1-Y')*log(1-H));
J=X'*(H-Y)/m;
a=0.05;%learning rate
lambda=10;%正则化系数
for i=1:10000
    sum=0;%正则化项
    Z=X*B;%simoid 自变量  m*1 维
    for j=1:m
        H(j,:)=1/(1+exp(-Z(j,:)));%sigmiod 函数
    end

    for j=1:dim
        sum=sum+B(j,:)*B(j,:);
    end
    EC(i,:)=lambda*sum/m;
    E(i,:)=(-1/m)*(Y'*log(H)+(1-Y')*log(1-H))+lambda*sum/m;%Loss Function
    J=X'*(H-Y)/m+lambda*B/m;%梯度
    B=B-a*J;%梯度迭代
end
disp('loss')
E(i)
figure(1);
plot(E);%绘制 loss 与迭代次数的关系图
figure(2);
plot(EC);%绘制正则化项与迭代次数的关系图
```

三、结果分析

在 matlab 中配置相应的文件格式和路径，运行上述程序。*Loss* 与迭代次数的关系图以及正则化项与迭代次数的关系图如图 8-(a)、（b）所示；且根据题目

所给测试数据，可以预测出用户点击广告的概率，再将概率数据导出，与用户 ID 一起生成预测文件（submission.csv）。



（a） *Loss* 与迭代次数的关系图　　（b） 正则化项与迭代次数的关系图

图 8 *Loss* 与正则化项关于迭代次数的关系图

从上图中可以明显看出，loss 随着迭代次数的增加，明显降低，正则化项随着迭代次数的增加，明显增加，符合预期。

本文再将预测文件（submission.csv）上传到 lintcode，观察模型的预测效果，结果如图 9 所示。

文件及描述	状态	分数
predictions.csv 21 分钟前	成功	0.44838
predictions.csv 23 分钟前	提交格式错误	0.00000
predictions.csv 26 分钟前	提交格式错误	0.00000
predictions.csv 29 分钟前	提交格式错误	0.00000

排名　　刷新

#	队伍名称	分数	成员	提交次数	最后提交时间
1	seagullbird	0.00000		2	2 年前
2	HaoliZhang	0.00000		2	2 年前
3	SinceJune	0.00000		4	2 年前
4	ysf	0.42957		24	1 年前
5	yaohualiu2018	0.43230		7	1 年前
6	levy0834	0.43304		5	8 个月前
7	768219451	0.43330		11	1 年前
8	yinccc	0.44461		0	1 年前
9	1405081701	0.44838		12	22 分钟前
10	wuwuwu123	0.44899		3	2 年前
11	shanshan13	0.45488		0	2 年前

图 9 模型预测结果

从上图中可以看出，由于之前几次的文件提交格式设置有误，系统报错，待调整后再次上传，可以得到最终的分数为 0.44838，然后观察成绩排名，排名第 9 位（截至到作业提交前，共 63 组），排名比较靠前，说明该模型的预测效果较好，基本可以实现对用户点击广告的预测。

四、总结与致谢

本次作业是在考试结束很长时间后才开始写的，由于时间过去了很久，很多的知识点都已模糊，一开始接触 logistic 回归时，觉得一头雾水，对于问题觉得束手无策，在经过复习、查阅资料以及阅读文献后，渐渐地把学过的知识捡了起来，并通过本次大作业进一步加深了印象，感觉受益匪浅。

在这次大作业中，遇到的一个比较棘手的问题是对数据的处理，一开始，我只选择了前面 13 个连续特征进行仿真，但是缺省值很多，于是联想到上课老师所讲过的用平均值填充的方法，来对缺省值进行填补，发现结果还可以，之后，根据课上所学过的知识，尝试用随机森林法进行迭代填充，从而更加接近真实值，最后，验证出的结果确实更好了。这一过程，不仅加深了我对之前所学理论知识的理解，更通过自身实践加以证实，使得记忆更加深刻了

最后，非常感谢赵老师的细心教导与认真讲解，在不到十周的学习时间里，尽其所能地让我们多学一些知识，并且给予我们很多今后科研和工作道路上的宝贵建议，感谢赵老师的辛苦付出。