





知



首发于  
linger的脑洞

写文章

登录



linger liu · 2 年前

用户画像是啥？听起来很高大上的，其实你最熟悉不过了。你的性别，年龄，喜好等等这些都是用户画像的维度。迅雷的产品总监blues认为，用户画像分析的维度，可以按照人口属性和产品行为属性进行综合分析，

人口属性：地域、年龄、性别、文化、职业、收入、生活习惯、消费习惯等；

产品行为：产品类别、活跃频率、产品喜好、产品驱动、使用习惯、产品消费等；

那么问题又来了，互联网公司利用用户画像做啥呢？主要是个性化营销和精准广告。

假如一个电商网站知道你是一个妹子，那么给你推荐女性的商品，你更可能购买。

假如你是一个年轻的小伙子，在视频网站上观看英超比赛，此时给你投放一个体育用品的广告，尽管你不一定会点击甚至产生消费行为，但是总比给你投放一个妇女用品的广告的效果要好。

好了，吹牛逼到此为止。下面开始从技术层面谈谈如何实践用户画像建模。

每个公司拥有的用户信息不太一致，有些公司有用户的真实信息（比如年龄），这种情况很可能不需要建模预测了，本文着重讨论的是我们从用户行为上建模预测各种用户画像维度。

说到预测，自然就会想到机器学习的分类和回归算法（贝叶斯，决策树，逻辑回归，支持向量机等）。对，我们就是采取这些有监督的学习方法，从标注好的训练数据学习到一个预测模

知

首发于  
linger的脑洞

写文章

[登录](#)

从中文姓名预测性别说起

告诉我一个名字，让我猜猜是男是女。哈哈，多多少少有点算命的味道。

首先，有监督的学习方法，就需要这样一批标注数据：大量的人名，以及其性别。

下面是某网站的数据，

姓名和身份证	性别	年龄	所属地区
鲍俊豪 [REDACTED]	男	29	宁夏回族自治区
尤康裕 [REDACTED]	男	33	江苏省
柏懿轩 [REDACTED]	男	41	福建省
彭立诚 [REDACTED]	男	39	江苏省
周豪健 [REDACTED]	男	39	辽宁省
戚雄强 [REDACTED]	男	36	台湾省
华朗滔 [REDACTED]	男	27	云南省
苗健柏 [REDACTED]	男	31	湖南省

经过我的验证，上面的是真实的身份证数据，所以数据绝对靠谱和真实。

于是，从上面爬了几百万条的数据作为我们的训练集和测试集。

```
mysql> select * from Chinese_name_gender limit 100;
+-----+-----+-----+
| id_card_num | name | gender |
+-----+-----+-----+
| 11000019700110300X | 邬曼丽 | female |
| 110000197001103026 | 许虹英 | female |
| 110000197001103058 | 史远翔 | male |
| 110000197001103009 | 褚欣笑 | female |
| 110000197001103073 | 俞明旭 | male |
| 110000197001103071 | 苗心怡 | male |
| 110000197001103090 | 姜碧菡 | female |
| 110000197001103003 | 岑圣杰 | male |
```

其中，400w当做训练样本，100w当做测试样本。

分类算法：贝叶斯

特征提取流程：

1 根据姓氏辞典把姓氏去掉，留下不带姓氏的名字；

2 特征有三个维度，分别用X1,X2,X3(=X1X2)表示。

如果是单字名，则X1为空格，X2为单字名，X3就是前两者拼接X1X2，

比如郭靖，X1=" ",X2="靖"，X3=" 靖"。

比如黄药师，则X1="药"，X2="师"，X3="药师"

如果是三个字以上，则只保留最后两个字，当做双字名处理。

原始贝叶斯公式

$$P(B_i|A) = \frac{P(B_i)P(A|B_i)}{\sum_{j=1}^n P(B_j)P(A|B_j)}$$

工程实现中，上面公式的分母可以去掉。

所以对于名字X1X2,

$$P(\text{男}|X1X2)=P(\text{男})*P(X1|\text{男})*P(X2|\text{男})*P(X1X2|\text{男})$$

$$P(\text{女}|X1X2)=P(\text{女})*P(X1|\text{女})*P(X2|\text{女})*P(X1X2|\text{女})$$

特别注意的是，P(X1|男)表示的是训练样本中，男性用户中，名字第一个字出现X1的概率，如果第二个字出现X1，不算在这里。

举个例子，如何判断谢霆锋是男是女？

$$P(\text{男}|\text{霆锋})=P(\text{男})*P(\text{霆}|\text{男})*P(\text{锋}|\text{男})*P(\text{霆锋}|\text{男})$$

$$P(\text{女}|\text{霆锋})=P(\text{女})*P(\text{霆}|\text{女})*P(\text{锋}|\text{女})*P(\text{霆锋}|\text{女})$$

知



首发于  
linger的脑洞

写文章

登录

$P(\text{锋}|\text{男})$  表示的是训练样本中，男性用户第二个字出现锋的概率，

$P(\text{霆锋}|\text{男})$  表示的是训练样本中，男性用户第一个字是霆且第二个字是锋字的概率。

工程实现中，在预测阶段，可能会遇到一些特征在训练样本中没有，则需要做一下平滑（比如分子加一个很小的值），不然男女概率都为0，无法预测。

关于这个模型，特征提取和采取贝叶斯模型，是参考了《基于中文人名用字特征的性别判定方法》这篇论文来做的。还有篇论文，是用条件随机场来做的，有兴趣的读者可以看看《基于条件随机场的中文人名性别识别研究》。

在实际应用中，这个模型适合于我们知道用户姓名但是不知道性别的情况，比如某电商网站，一般情况用户订单中填的收货人姓名都是真实的，注册信息中可能带有性别但是不靠谱可能是乱填的。

提到电商，不得不提淘宝，淘宝可是有用户的身份证信息的，但是据说淘宝会保存用户两个性别，一个是身份证信息的性别，一个是预测性别。也许是他们实践过，不同的业务场景，两个性别的效果不一样。

好了，再说说从用户行为来建模预测。不同的产品的用户行为信息不太一致，提取的特征以及方法也不一致。对于视频用户，提取的特征通常是这个视频信息，比如你看了《小时代》，我们就会给你打上一个“杨幂”的标签作为你其中一个特征。对于微博用户，提取的特征来自于你发的微博内容和关注的好友，比如你关注了tfboys，或者经常转发他们的相关的微博，我们就会给你打上“tfboys”和“小鲜肉”的特征。对于新闻资讯用户，提取的特征就是该新闻的主题，关键词等信息。提取的特征来自于结构化信息和非结构化信息。对于结构化信息（数据库表，xml,json等），容易抽取。但是对于非结构化信息，就需要自然语言处理和文本挖掘的方式来

知



首发于  
linger的脑洞

写文章

登录



对于抽取的这些信息，就可以当做用户的基本行为特征，其实也可以当做一部分用户画像的维度了。

然后，我们如何利用这些基本行为特征，来进行其他用户画像维度的预测呢？

还是采取分类的算法。对于有监督学习，当然需要男女，年龄等标注数据了。可以跟第三方公司进行数据合作，也可以人工标注。通常，可以跟第三方广告公司进行数据交换，通过cookie-mapping方式进行用户关联。

好了，特征和标注数据都有了，那么可以建模了。分类算法一大堆，贝叶斯，逻辑回归，svm，gbdt，random forest，神经网络等等。开源的算法库也有很多，后续会介绍liblinear,xgboost,sklearn。

然而，经过实践的工程师都知道，算法不是万能的。一个好的模型，除了算法本身，数据质量，数据预处理，特征工程等等这些都很重要。且外，还需要根据业务情况，进行随机应变。

下面分享一些工作中用到的奇技淫巧。

互联网数据，并不像学术研究中的数据那么完美，而是“杂乱脏”。抛开业务不说，仅从数据上看，标注数据有误，特征数据稀疏，类别不平衡等等。总之由于种种问题，可能导致我们模型不是那么准确。一种有效的思路，就是我们不预测全体用户，只预测我们有把握预测的，因为对于业务方来说，准确率相对于召回率，对他们更重要。朝着这种思路，也有不少尝试方案。比如分群预测，分群的方法可以从业务上分，也可以从算法上分。从业务上分，比如视频用户可以分为电影用户和电视用户等等。从算法上分，可以通过聚类的方法来分。分群完毕后，再对不同的群体进行建模，通过测试集预测，过滤掉准确率较低的群体，保留准确率较高的群体即可。还有种尝试方案，就是特征选择和用户筛选结合起来，特征选择的算法也有很多，比如

知



首发于  
linger的脑洞

写文章

登录



保留这个用户取决于这个用户是否拥有足够多的有分类能力的特征。这个用户过滤环节，训练测试阶段和预测阶段都要进行。

好了，talk is cheap,show me your code。下面开始介绍几个开源算法工具库。

下面三点是我挑选工具的标准：

1 支持类似libsvm格式的稀疏特征输入

2 支持百万级别训练数据

3 轻量级，快

libsvm和liblinear

在特征维度是几万以上级别的前提下，

libsvm只能训练几万条样本。

而liblinear对于百万级别的样本数量，速度很快，

千万级别有点吃力。

liblinear对于libsvm的缺点就是，不能使用核函数，libsvm也正是这点所以耗性能。

数据文件格式：标准的libsvm格式，每一行都是

label index1:value1 index2:value2



```

0 143:0.4472135954999579 706:0.4472135954999579 779:0.4472135954999579 809:0.4472135954999579 814:0.4472135954999579
0 87:0.24253562503633297 88:0.24253562503633297 89:0.24253562503633297 90:0.24253562503633297 92:0.24253562503633297 93:0.
297 98:0.24253562503633297 99:0.24253562503633297 100:0.24253562503633297 101:0.24253562503633297 102:0.24253562503633297
0 87:0.2773500981126146 88:0.2773500981126146 89:0.2773500981126146 90:0.2773500981126146 92:0.2773500981126146 93:0.27735
.2773500981126146 99:0.2773500981126146 100:0.2773500981126146 101:0.2773500981126146 102:0.2773500981126146
0 190:0.4082482904638631 192:0.4082482904638631 198:0.4082482904638631 199:0.4082482904638631 212:0.4082482904638631 235:0
0 87:0.2773500981126146 88:0.2773500981126146 89:0.2773500981126146 90:0.2773500981126146 92:0.2773500981126146 93:0.27735
.2773500981126146 99:0.2773500981126146 100:0.2773500981126146 101:0.2773500981126146 102:0.2773500981126146
0 72:0.3333333333333333 73:0.3333333333333333 74:0.3333333333333333 75:0.3333333333333333 77:0.3333333333333333 78:0.33333
0.3333333333333333
0 87:0.31234752377721214 137:0.31234752377721214 143:0.31234752377721214 266:0.31234752377721214 457:0.4685212856658182 49
377721214 581:0.31234752377721214

```

liblinear最简单的训练和测试方式：

train train.txt model.txt

predict test.txt model.txt predict.txt

使用liblinear之前，

建议进行L2范式归一化，实现简单，可以加快训练速度，可能会提高准确率。

$$\begin{aligned}
 l = \text{norm}(x') &= \frac{\sqrt{x_1^2 + x_2^2 + \dots + x_n^2}}{\text{norm}(x)} \\
 &= \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_n^2}{\text{norm}(x)^2}} \\
 &= \sqrt{\left(\frac{x_1}{\text{norm}(x)}\right)^2 + \left(\frac{x_2}{\text{norm}(x)}\right)^2 + \dots + \left(\frac{x_n}{\text{norm}(x)}\right)^2} \\
 &= \sqrt{x_1'^2 + x_2'^2 + \dots + x_n'^2} \\
 \text{即: } x_i' &= \frac{x_i}{\text{norm}(x)}
 \end{aligned}$$

提到该工具，不得不秀一下这个工具的作者陈天奇大神。



简介：陈天奇，华盛顿大学计算机系博士生，研究方向为大规模机器学习。他曾获得KDD CUP 2012 Track 1第一名，并开发了SVDFeature, XGBoost, cxxnet等著名机器学习工具，是Distributed (Deep) Machine Learning Common的发起人之一。

xgboost是gbdt算法的实现，可以做回归，分类，和排序。支持各种语言调用，支持单机和分

知



首发于  
linger的脑洞

写文章

登录

项目主页

[dmlc/xgboost · GitHub](#)

单机版的python调用例子：

```
linger@ubuntu:/mnt/sda4/data/ad_gender_letv_all$ cat xgb.py
#!/usr/bin/python
import numpy as np
import xgboost as xgb

dtrain = xgb.DMatrix('train.txt')
dtest = xgb.DMatrix('test.txt')

# specify parameters via map, definition are same as c++ version
param = {'max_depth':12, 'eta':0.1, 'silent':0, 'objective':'binary:logistic', 'min_child_weight':3, 'gamma':7 }

# specify validations set to watch performance
watchlist = [(dtest,'eval'), (dtrain,'train')]
num_round = 77
bst = xgb.train(param, dtrain, num_round, watchlist)

# this is prediction
preds = bst.predict(dtest)
labels = dtest.get_label()
print ('error=%f' % ( sum(1 for i in range(len(preds)) if int(preds[i]>0.5)!=labels[i]) /float(len(preds))))
print ('correct=%f' % ( sum(1 for i in range(len(preds)) if int(preds[i]>0.5)==labels[i]) /float(len(preds))))
```

《xgboost快速入门》

[xgboost快速入门](#)

sklearn

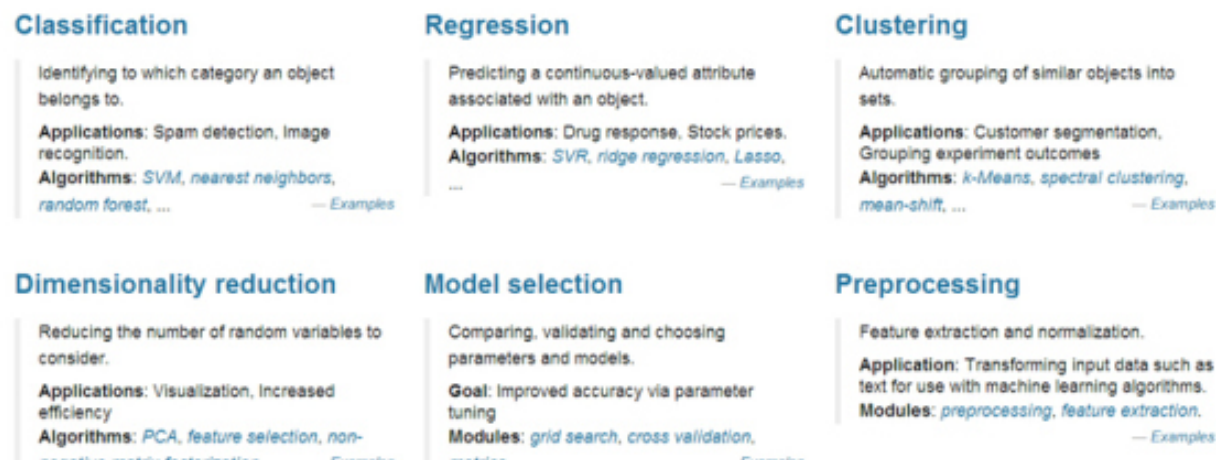
知



首发于  
linger的脑洞

写文章

登录



涵盖数据预处理，降维，分类，回归，聚类，模型选择。

sklearn也支持libsvm的数据格式，

但是某些算法不一定支持，

比如1.5版本的随机森林不支持稀疏矩阵，1.6就支持了。

1.6版本的gbdt也不支持稀疏矩阵。

kmeans的调用例子



```
from time import time
import numpy as np
import matplotlib.pyplot as plt
from sklearn.externals import joblib

from sklearn import metrics
from sklearn.cluster import KMeans
from sklearn.datasets import load_digits
from sklearn.decomposition import PCA
from sklearn.preprocessing import scale
from sklearn import preprocessing
from sklearn.externals.joblib import Memory
from sklearn.datasets import load_svmlight_file
mem = Memory("./mycache")

@mem.cache
def get_data():
    data = load_svmlight_file("train.txt")
    return data[0], data[1]

X, y = get_data()
#X = scale(X,with_mean=False)

t0 = time()
kmeans = KMeans(init='k-means++', n_clusters=7, n_init=7)
kmeans.fit(X)

joblib.dump(kmeans, './cluster/cluster.model')

print time()-t0

print kmeans.labels_[0:3]
```

《sklearn特征选择和分类模型》

知



首发于  
linger的脑洞

写文章

登录

好了，经历了用户画像建模的这段经历，的确体验到算法并不是万能的。所以，我们出了把精力学习那些神奇的算法原理，还需要在数据分析，数据处理，特征工程甚至业务知识方面下功夫。特别是特征工程，在互联网中是永恒的主题。百度凤巢在很久之前，一直沿用人肉特征工程+线性模型的方法，也是近几年才引用深度学习来做特征。最近有篇Facebook的论文，是用gbdt提特征输入给lr预测，有时间我也会学习一下。

好了，晚安。

2015年11月25号夜


本文作者:linger

本文链接：[用户画像建模：方法与工具](#)

转载须注明出处

欢迎关注公众号：数据挖掘菜鸟【公众号ID：data\_bird】



 数据挖掘菜鸟

知



首发于  
linger的脑洞

 写文章

登录



关注大数据，数据挖掘，机器学习，深度学习

「真诚赞赏，手留余香」

赞赏

还没有人赞赏，快来当第一个赞赏的人吧！

用户画像

数据挖掘

机器学习

用户行为数据



166

☆ 收藏

📄 分享

⚠️ 举报



...

文章被以下专栏收录



linger的脑洞  
漫谈数据挖掘

进入专栏

知



首发于  
linger的脑洞

📄+ 写文章

登录

写下你的评论...



**瑞恩-Rayn**

求数据来源???

2 年前



**linger liu** (作者) 回复 **瑞恩-Rayn**

查看对话

自己公司产品的用户数据啊，这个因公司而异的

2 年前



**Chong Pang**

我先通过贝叶斯对用户画像，得到用户模型。再聚类，得到具有该用户画像特征的用户聚类  
这样有问题吗？聚类和分类有什么区别？谢谢

1 年前



**凯文**

不明觉厉

1 年前

**周小涛**



首发于  
**linger的脑洞**

知

写文章

登录

1 年前

**徐小磊** 回复 **Chong Pang**[查看对话](#)

聚类：未知具体的类别，有可能收敛为1个或3个或多个类别；分类，已知具体的类别，例如，已知有男人和女人两类，把结果丢进要么男人，要么女人。

1 年前

1 赞

**Steven Liu** 回复 **Chong Pang**[查看对话](#)

你先搞清楚什么是有监督和无监督吧

10 个月前

**linger liu**（作者） 回复 **周小涛**[查看对话](#)

Practical lessons from predicting clicks on ads at facebook

10 个月前

**miller**

$P(\text{男}|X_1X_2)=P(\text{男})\cdot P(X_1|\text{男})\cdot P(X_2|\text{男})\cdot P(X_1X_2|\text{男})$ ,这样对吗？

9 个月前

**陈辉**

mark

8 个月前

首发于  
**linger的脑洞**

知

[+ 写文章](#)[登录](#)

## 推荐阅读



### 从用户转化来谈谈数据化运营：拉新，留存，回流等

近段时间开会经常提到用户运营，从同事口中听到一些关于运营手段和技术需求。我总... [查看全文](#) >

linger liu · 2 年前 · 发表于 linger的脑洞



### 人工特征工程的意义

最近在做用户画像，比如性别等的预测模型。由于准确率不高，一直在尝试各种方法，包括加大训... [查看全文](#) >

linger liu · 2 年前 · 发表于 linger的脑洞



### 澳大利亚铭德律师事务所关于FIRB审批最新实务发展的讲座PPT

很高兴和澳大利亚铭德国际律师事务所合作举办“跨境并购交易实务发



Zhang Leslie · 11 天前 · 编辑精选 · 发表于 UncleLeslie的看法



## 出轨不会净身出户，但这个绝对可以！

今天要讲一个净身出户的故事。如果不能挑战你们神经，算我输。这一对夫妻，也算是门当户对了... [查看全文](#) >

吴杰臻律师 · 2 天前 · 编辑精选 · 发表于 离婚大师说