旧金山犯罪分类问题技术报告

SunshineBot

***Abstract*—旧金山犯罪分类问题是在Kaggle上举办的一个在线比赛。该比赛数据来自旧金山所有街区2003年到2015年的犯罪报告，提供了犯罪的时间和地点等信息，要求参赛者预测犯罪的类别。我分析了不同特征与犯罪数量及类型之间的关系，并进行了不同的处理，包括高斯混合模型（Gaussian Mixture Model, GMM）、谷歌的word2vec等，最后使用了DMLC的XGBoost和微软的LightGBM分类器进行预测。**

***Index Terms*—多分类，LightGBM，Kaggle**

I. 引言

从1934年到1963年，旧金山因为恶魔岛上臭名昭著的犯罪事件而闻名于世。如今这座城市因其技术领域而闻名，但是随着贫富差距变大、住房短缺、以及大量电子玩具的泛滥，这座海湾城市并不缺少犯罪。Kaggle的比赛数据集来自旧金山所有街区2003年到2015年的犯罪报告，主要提供了时间、街区、经纬度等信息，我需要从这些信息中预测39种犯罪类别发生的可能性。

数据集中主要包括了时间和地理信息。对于一个完整的时间信息，我将其按照年月日等单位进行拆分，并分析不同时段犯罪数量和种类之间的关系，以此对时间特征进行处理。地理信息包括街区、地址、经纬度等多种信息。我采用了聚类、one-hot编码、二值装箱、词向量多种方法进行处理。

最终我的方法评分2.23914，在Kaggle公开榜单中排名前4%(82/2335)。

II. 特征分析

该数据集包含来自SFPD犯罪事件报告系统的事件。数据范围从2003年1月1日到2015年5月13日。训练集和测试集每周轮换，意味着第1、3、5、7等周属于测试集，第2、4、6、8等周属于训练集。数据集中包含以下特征:

- **Dates** 犯罪事件发生的时间，格式：2003-11-23 08:23:23
- **Category** 犯罪事件的类型（仅出现在训练集），也是要预测的特征
- **Descript** 犯罪事件的详细描述（仅出现在训练集）
- **DayOfWeek** 当天是星期几
- **PdDistrict** 所管辖的警察局所在的地区
- **Resolution** 犯罪事件如何处理（仅出现在训练集）
- **Address** 犯罪事件发生的大致地址
- **X** 经度
- **Y** 纬度

此外，测试集中还有一个特征:

- **Id** 需要预测的犯罪事件的编号

A. 时间特征处理

时间特征包含Dates和DayOfWeek两个特征。

Dates特征拆分为年、月、日、时、分5个特征，考虑到秒对犯罪的发生应当没有影响，我丢弃了秒这个特征。

我对各个时段发生犯罪事件的几率进行了统计，发现对于小时和分这两个特征，犯罪发生的概率呈现出非常明显的差异。对于小时而言，如图2，从半夜1点开始，犯罪数量大幅下降，直到上午8点才逐渐恢复到一个较高的水平，到中午12点时达到了一个顶峰，下午时稍微下跌，到18时达到了另一个顶峰，此后逐渐下降。因此我将小时特征拆分为四类。但是在后续的测试中我发现，这个特征反而降低了分类的准确度。因此最终我去掉了这个特征。对于分钟而言，如图1我发现整点时（分钟数为0）的犯罪数量约占总犯罪数量的30%，在半点时（分钟数为30）的犯罪数量约占总犯罪数量的14%，在15和45分的时候犯罪数量也很高，在分钟数为5的倍数时的犯罪数量为其它时段的5倍左右。此外，分钟数为1时的犯罪数量也明显较高，因此我将分钟数为0、1、15、30、45的样本分为一类，其它样本分为一类。

DayOfWeek特征我将它处理为有序的数字，从周一到周日依次为0到6.

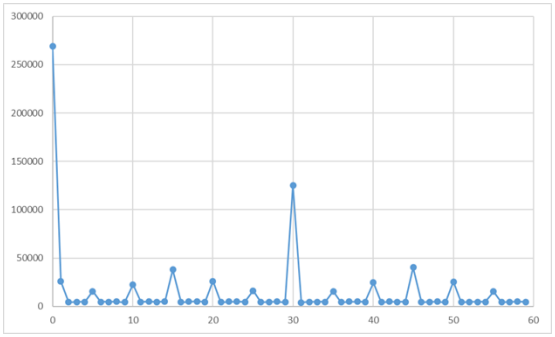

Fig. 1. 不同分钟的犯罪数量

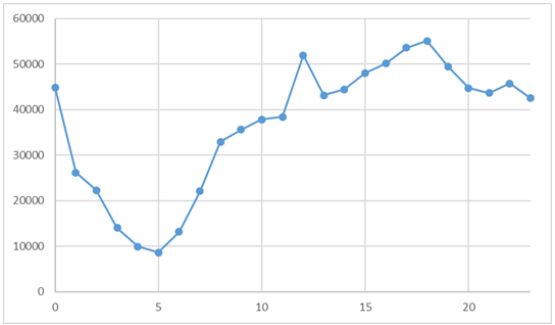

Fig. 2. 不同小时的犯罪数量

B. 区域特征处理

PdDistrict特征表示应该管辖该犯罪事件的警察局所在的地区，由于仅有十个区域，我简单的进行了one-hot处理。

C. 地址特征处理

地址特征非常的难处理，因为它是字符串类型的，数量也非常多（约23000个），如果直接进行one-hot处理会消耗大量的内存。经过与他人讨论，我尝试将出现次数少于一定数量的地址全部归入others类别，得到一千多个地址，但是在测试中发现分类准确度反而会降低。于是我在前期调整模型参数的时候并没有使用地址特征。

地址特征主要分为两类，一种是两条道路的交汇点，值中包含"/"，另一种是街道上的一个街区，值中包含"Block of"，相比于粗粒度的区域特征和由二维点构成的经纬度特征，地址特征中显然也蕴含着重要的信息。考虑到地址一般由5个左右的单词组成，单词之间既有差异又有相同的部分，我试图挖掘单词中蕴含的词义信息。最终我采用谷歌发布的开源工具word2vec来处理地址特征，试图寻找不同地址特征之间的相似性和差异性。我将每个样本（包含测试集）的地址信息作为一个句子，整个训练集和测试集的所有地址信息作为文档，使用word2vec训练成了100维的词向量。然后将每个样本的地址信息拆分成单词，将他们的词向量相加作为地址特征，放入分类器中进行训练。令我感到意外的是，这样的处理有着非常明显的效果，说明word2vec可以有效的挖掘地址特征中包含的信息。

D. 经纬度特征

考虑到不同地理位置的犯罪类别应当不一样，我采用了高斯混合模型对经纬度进行了聚类处理，并对聚类的结果进行了one-hot处理。

III. 评价方法

提交的结果使用对分类对数损失函数。每个测试样本都有一个正确的类别，对每个样本，我要提交它发生每种犯罪的概率，计算公式如下：

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}),$$

其中N表示测试集中的样本数量，M表示类别标签的数量，log是自然对数，如果第i个样本的真实值是类别j，则$y_{ij}=1$，否则$y_{ij}=0$，p_{ij}是第i个样本属于第j类犯罪的概率。

对于每个样本，提交的个类别的概率之和不需要为1，因为在计算之前他们会被重新缩放。为了避免log函数的极值，预测概率会被公式$max(min(p, 1-10^{-15}), 10^{-15})$取代。

IV. 模型设计

在特征处理上，我对经纬度采用了高斯混合模型进行聚类，对地址信息采用了word2vec模型训练出词向量以提取特征。在分类训练上，我尝试了简单决策树、随机森林、Adaboost、XGBoost和LightGBM模型，决策树和随机森林效果较差，Adaboost效果尚可但是仍然不能令人满意，最终我采用了XGBoost和LightGBM模型。

A. 高斯混合模型

高斯密度函数估计是一种参数化模型。高斯混合模型用于对经纬度信息进行聚类，根据高斯概率密度函数（Probability Density Function, PDF）参数不同，每一个高斯模型可以看作一种类别，输入一个样本x，即可通过PDF计算其值，然后通过一个阈值来判断该样本是否属于高斯模型。GMM由于由多个单高斯模型混合，划分更为精细，适用于多类别的划分，可以应用于复杂对象建模。

如表I,我尝试了不同的类别数量，最终确定的GMM参数如表II所示：

TABLE I
GMM不同类别数量的分类效果

类别数量	训练集评分	测试集评分
110	2.147999	2.353570
120	2.150720	2.351660
130	2.153782	2.350373
140	2.154448	2.348725
150	2.165074	2.355563
160	2.167476	2.356295

备注：采用LightGBM分类器，参数：n_estimators=300, learning_rate=0.1, num_leaves=128

TABLE II
GMM参数

参数	值
n_components	140
covariance	diag

B. *word2vec*模型

Word2vec是谷歌在2013年开源的一款计算词向量的工具，它使用了浅层的神经网络，可以在百万数量级的词典和上亿的数据集上进行高效地训练，它得到的词向量不仅可以很好的衡量词语之间的相似性，还包含了语义信息，引起了学术界和工业界的关注。在旧金山犯罪分类这个问题中，我使用word2vec对地址特征进行了训练，分别尝试了50维和100维的词向量，均获得了比较好的提升效果。最终确定的Word2vec的参数如表III所示。

TABLE III
WORD2VEC参数

参数	值
size	100
min_count	1
epochs	200

C. *XGBoost*模型

XGBoost是一个优化的分布式梯度增强库，旨在高效，灵活和便携。它在梯度增强框架下实现机器学习算法。XGBoost提供了一个并行树增强（也称为GBDT，GBM），以快速和准确的方式解决了许多数据科学问题。相同的代码运行在主要的分布式环境

（Hadoop，SGE，MPI）上，可以解决超过数十亿个例子的问题。在Kaggle的分类比赛中，XGBoost是在越来越多的比赛中获得冠军，相比其它的机器学习分类算法，XGBoost有着更好的分类效果。但是它的训练速度仍然较慢，这让参数调节变得困难。

在这个模型中，我发现特征数量较少可以获得更高的结果，因此我将经纬度聚类成20类。并且，我没有对地址特征进行词嵌入处理，而是单纯删去了该特征。具体参数如表IV

TABLE IV
XGBOOST参数

参数	值
max_depth	8
n_estimators	50
objective	multi-softprob
eval-metric	mlogloss

由于XGBoost训练速度较慢，后期我使用了LightGBM作为分类器，因此XGBoost并没有取得非常好的成绩。最终评分为2.43708。

*D. LightGBM*模型

LightGBM是微软开发的分布式梯度提升框架，同样使用基于树的学习算法。相比XGBoost，LightGBM在不降低准确率的情况下，可以获得更快的训练速度，降低内存使用量，还支持GPU加速。公共数据集上的比较实验表明，LightGBM在效率和准确性方面都可以超越现有的增强框架，并且内存消耗也大大降低。并行实验表明，LightGBM可以通过在特定环境下使用多台机器进行训练来实现线性加速。如表V，我发现使用较小的学习率和较多的树，分类的准确率会慢慢提升，但是会降低训练速度。同时为了防止过拟合，我对样本进行了下采样。树深度选择了适中的数值，叶子节点个数依据树深度的数值适当调大，在过拟合和欠拟合之间寻找一个适当的平衡点。经过多次尝试，GMM聚类的类别数为140类，word2vec训练的词向量维数为100维可以取得较好的效果，LightGBM分类器的参数如表VI所示。

TABLE V
LIGHTGBM不同N_ESTIMATORS下的分类效果

n_estimators	训练集评分	测试集评分
50	2.422454	2.440458
100	2.369238	2.406143
150	2.344883	2.401131
200	2.324805	2.399340
250	2.306737	2.398466
300	2.290043	2.397985
350	2.274443	2.397938

备注：参数：learning_rate=0.1, num_leaves=31

最终我的模型评分为2.23914，在Kaggle公开榜单中排名前4%(82/2335)。

E. 特征重要性

在使用LightGBM分类器取得的最优结果中，我输出了各项特征在分类模型中的重要性，值越大说明该特征在分类的过程中越重要。从表VII可以看出，时间特征的重要性远远高于地址特征，我猜测是因为时间特征的维数较少，单个特征蕴含的信息较多，因而重要性较高，而地址特征维数较高，单个特征中蕴含的信息有限，导致重要性较低。从表VIII可以看出，地理特征的重要性总和远高于时间特征的重要性总和，说明地理特征在判断犯罪类别的过程中起到了更加重要的作用。

TABLE VI
LIGHTGBM参数

参数	值
max_depth	8
n_estimators	350
objective	multiclass
learning_rate	0.05
subsample	0.6
num_leaves	128
eval-metric	mlogloss

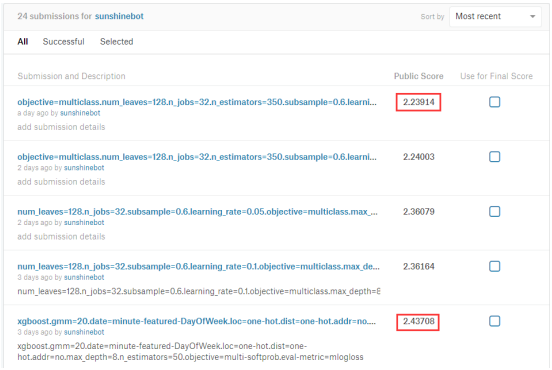

Fig. 3. Kaggle成绩

TABLE VII
LIGHTGBM模型特征重要性

特征	gini importance
hour	117475
date	110835
minute	93730
year	93412
month	74941
DayOfWeek	56191
minute_feature	22620
addr_0	13122
addr_2	12329
addr_51	12521
addr_30	12150
addr_25	11883
addr_41	11851
addr_32	11597
addr_60	11595

备注：addr_n表示地址特征训练得到的词向量的第n维

V. 结论

在本文中，我对特征采取了包括聚类、词向量等多种方式进行处理，然后使用了XGBoost和LightGBM两个分类器进行训练，最后在旧金山分类问题取得了比较好的效

TABLE VIII
LIGHTGBM模型时间和地理特征重要性总和

特征	gini importance
time_feature	569204
location_feature	1118162

果，评分2.23914，排名前4%（82/2335），这样的成绩也让我收到了鼓舞。之前我有想法参加Kaggle的比赛，但是没有付诸行动。这次的作业让我感受到我对数据比赛也有浓厚的兴趣，后续我可能也会参加一些正在举办的比赛。