[风控算法大赛解决方案]

by 陈靖 朱治亮 周耀 赵蕊 黄伟鹏

一、解决方案概述

1.1 项目介绍与问题分析

拍拍贷“魔镜风控系统”从平均 400 个数据维度评估用户当前的信用状态，给每个借款人打出当前状态的信用分，在此基础上再结合新发标的信息，打出对于每个标的 6 个月内逾期率的预测，为投资人提供关键的决策依据。本次竞赛目标是根据用户历史行为数据来预测用户在未来6个月内是否会逾期还款的概率。

问题转换成2分类问题，评估指标为AUC，从Master，LogInfo，UpdateInfo表中构建 特征，考虑评估指标为 AUC，其本质是排序优化问题，所以我们在模型顶层融合也使用基于排序优化的RANK_AVG融合方法。

1.2 项目总体思路

本文**首先从数据清洗**开始，介绍我们对缺失值的多维度处理、对离群点的剔除方法以及 对字符、空格等的处理；**其次进行特征工程**，包括对地理位置信息的特征构建、成交时间特 征、类别特征编码、组合特征构建、UpdateInfo 和 LogInfo 表的特征提取等；**再次进行特 征选择**，我们采用了 xgboost，xgboost 的训练过程即对特征重要性的排序过程；**然后处理 类别的不平衡度**，由于赛题数据出现了类别不平衡的情况，我们采用了代价敏感学习和过采 样两种方法，重点介绍我们所使用的过采样方法；**最后一部分是模型设计与分析**，我们采用 了工业界广泛应用的逻辑回归模型、数据挖掘比赛大杀器 xgboost，创新性地探索了 large-scale svm 的方法在本赛题上的应用，取得了不错的效果，此外还介绍了模型融合方 法。

二、数据清洗

2.1 缺失值的多维度处理

在征信领域，用户信息的完善程度可能会影响该用户的信用评级。一个信息完善程度为

100%的用户比起完善程度为 50%的用户，会更加容易审核通过并得到借款。从这一点出发，我们对缺失值进行了多维度的分析和处理：

按列（属性）统计缺失值个数，进一步得到各列的缺失比率，下图（图 1）显示了含有 缺失值的属性和相应的缺失比率：

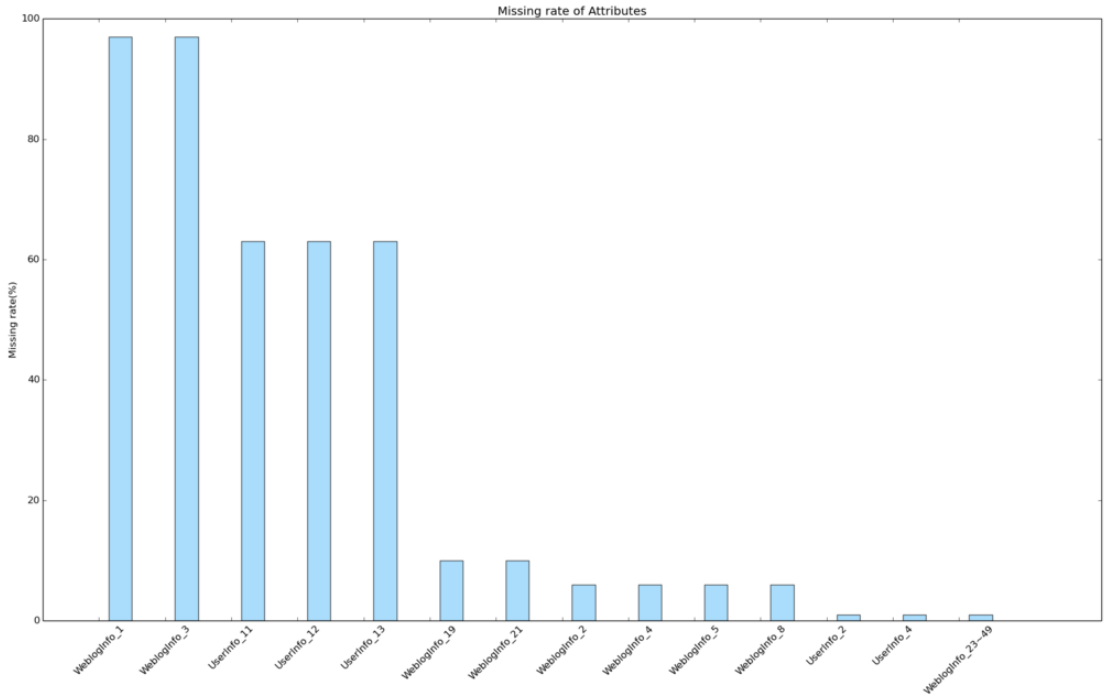

图 1.属性缺失比率

WeblogInfo_1 和 WeblogInfo_3 的缺失值比率为 97%，这两列属性基本不携带有用的信息，直接剔除。UserInfo_11、UserInfo_12 和 UserInfo_13 的缺失值比率为 63%，这三列属 性是类别型的，可以将缺失值用-1 填充，相当于“是否缺失”当成另一种类别。其他缺失 值比率较小的数值型属性则用中值填充。

按行统计每个样本的属性缺失值个数，将缺失值个数从小到大排序，以序号为横坐标，缺失值个数为纵坐标，画出如下散点图（图2）：

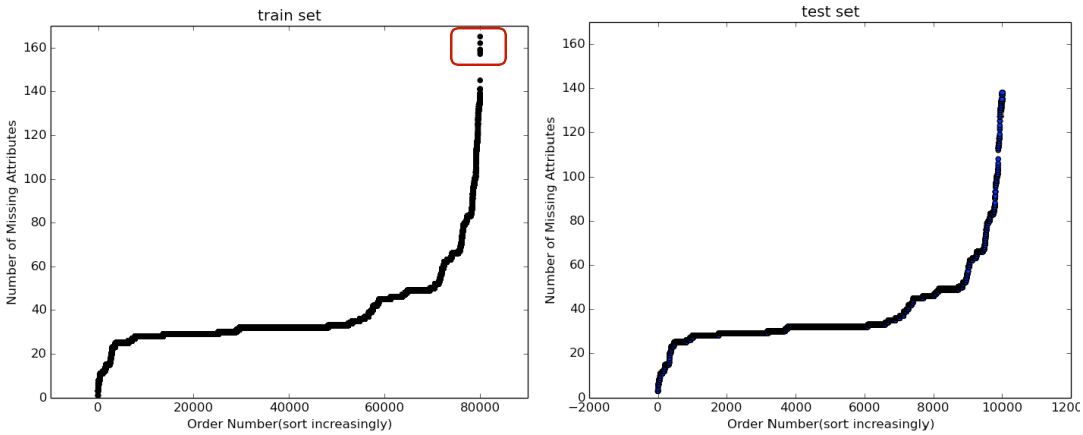

图 2.样本属性缺失个数

对比 trainset 和 testset 上的样本的属性缺失值个数，可以发现其分布基本一致，但是 trainset 上出现了几个缺失值个数特别多的样本（红框区域内），这几个样本可以认为是 离群点，将其剔除。

另外，缺失值个数可以作为一个特征，衡量用户信息的完善程度。

2.2 剔除常变量

原始数据中有 190维数值型特征，通过计算每个数值型特征的标准差，剔除部分变化很小的特征，下表（表1）列出的15个特征是标准差接近于0的，我们剔除了这15维特征。

表 1.剔除数值特征标准差

属性	标准差	属性	标准差	属性	标准差
WeblogInfo_10	0.0707	WeblogInfo_41	0.0212	WeblogInfo_49	0.0071
WeblogInfo_23	0.0939	WeblogInfo_43	0.0372	WeblogInfo_52	0.0512
WeblogInfo_31	0.0828	WeblogInfo_44	0.0166	WeblogInfo_54	0.0946
WeblogInfo_32	0.0834	WeblogInfo_46	0.0290	WeblogInfo_55	0.0331
WeblogInfo_40	0.0666	WeblogInfo_47	0.0401	WeblogInfo_58	0.0609

2.3 离群点剔除

在样本空间中与其他样本点的一般行为或特征不一致的点称为离群点，考虑到离群点的异常特征可能是多维度的组合，我们通过分析样本属性的缺失值个数，剔除了极少量的离群点（见3.1节）。

此外，我们还采用了另外一种简单有效的方法：在原始数据上训练 xgboost，用得到的xgb模型输出特征的重要性，取最重要的前20个特征（如图3所示），统计每个样本在这20个特征上的缺失值个数，将缺失值个数大于10的样本作为离群点。

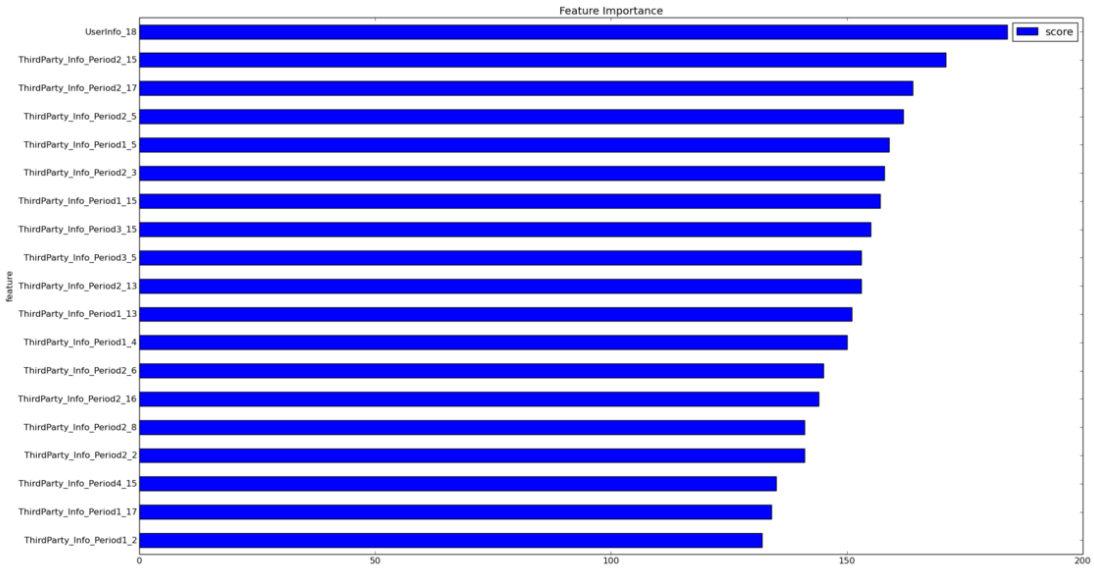

图 3. Xgb特征重要性

通过这个方法，剔除了 400多个样本。这些样本在重要特征上的取值是缺失的，会使得模型学习变得困难，从这个角度来说，它们可以看成是离群点，应剔除掉。

2.4 其他处理

（1）字符大小写转换

Userupdate_Info表中的UserupdateInfo1字段，属性取值为英文字符，包含了大小写，如"_QQ"和"_qQ"，很明显是同一种取值，我们将所有字符统一转换为小写。

（2）空格符处理

Master表中UserInfo_9字段的取值包含了空格字符，如“中国移动”和“中国移动 ”；它们是同一种取值，需要将空格符去除。

（3）城市名处理

UserInfo_8 包含有“重庆”、“重庆市”等取值，它们实际上是同一个城市，需要把 字符中的“市”全部去掉。去掉“市”之后，城市数由600多下降到400多。

三、特征工程

3.1 地理位置的处理

对地理位置信息（类别型变量）最简单的处理方式是独热编码（one-hot encoding），但是这样会得到很高维的稀疏特征，影响模型的学习，我们在独热编码的基础上，做了特征选择。下面介绍具体的方法。

赛题数据提供了用户的地理位置信息，包括 7 个字段：UserInfo_2、UserInfo_4、

UserInfo_7、UserInfo_8、UserInfo_19、UserInfo_20，其中 UserInfo_7 和 UserInfo_19 是省份信息，其余为城市信息。我们统计了每个省份和城市的违约率，下图以 UserInfo_7 为例：

图 4.省份违约率可视化

图 5可视化了每个省份的违约率，颜色越深代表违约率越大，其中违约率最大的几个省份或直辖市为四川、湖南、湖北、吉林、天津、山东，如下图所示：

图 5.违约率突出省份可视化

因此我们可以构建 6个二值特征：“是否为四川省”、“是否为湖南省”....“是否为 山东省”，其取值为 0 或 1。其实这相当于对地理位置信息做了独热编码，然后保留其中有

判别性的某些列。这里UserInfo_7包含32种取值，编码后可以得到32维的稀疏特征，而 我们只保留其中的6维。

以上我们是通过人工的分析方法去构建二值特征，在处理省份信息时还是比较直观的，但是处理城市信息，比如 UserInfo_2，包含了 333 个城市，就没有那么直观了。为了得到 有判别性的二值特征，我们首先对UserInfo_2进行独热编码，得到333维的二值特征，然 后在这333维稀疏特征上训练xgb模型，再根据xgb输出的特征重要性刷选二值特征，以下 是选取到的部分二值特征（对应的城市）：“淮纺市”、“九江市”、“三门峡市”、“汕 头市”、“长春市”、“铁岭市”、“济南市”、“成都市”、“淄博市”、“牡丹江市”。

- **按城市等级合并** 类别型特征取值个数太多时，独热编码后得到太高维的稀疏特征，除了采用上面提到的

特征选择方法外，我们还使用了合并变量的方法。按照城市等级，将类别变量合并，例如一线城市北京、上海、广州、深圳合并，赋值为 1，同样地，二线城市合并为 2，三线城市合并为3。

- **经纬度特征的引入** 以上对地理位置信息的处理，都是基于类别型的，我们另外收集了各个城市的经纬度，

将城市名用经纬度替换，这样就可以将类别型的变量转化为数值型的变量，比如北京市，用经纬度（39.92,116.46）替换，得到北纬和东经两个数值型特征。加入经纬度后，线下的cross validation有千分位的提升。

- **城市特征向量化**

我们将城市特征里的城市计数，并取 Log，然后等值离散化到 6~10 个区间内。以下图为例，将 UserInfo_2 这个特征里面的 325 个城市离散为一个 6 维向量。向量“100000”表 示该城市位于第一个区间。线下的cross validation有千分位的提升。

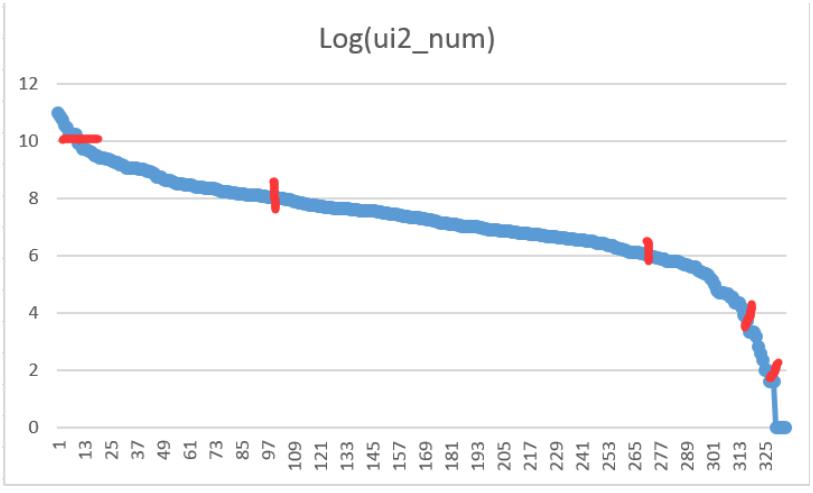

图 6.城市特征离散化

- **地理位置差异特征**

如图8所示，1,2,4,6列都是城市。那么我们构建一个城市差异的特征，比如diff_12 表示1，2列的城市是否相同。如此构建diff_12,diff_14,diff_16,diff_24,diff_26,diff_46这6个城市差异的特征。线下的cross validation有千分位的提升。

Idx	UserInfo_2	UserInfo_4	UserInfo_7	UserInfo_8	UserInfo_19	UserInfo_20
10005	广州	韶关	广东	广州	辽宁省	锦州市
10013	郴州	广州	广东	广州	湖南省	郴州市
10020	惠州	惠州	广东	惠州	四川省	广安市
10033	枣庄	枣庄	山东	枣庄	山东省	枣庄市
10035	深圳	南平	福建	南平	福建省	不详
10038	济宁	济宁	山东	济宁	山东省	济宁市
1004	连云港	连云港	江苏	连云港	江苏省	连云港市
10042	德州	德州	山东	滨州	山东省	德州市
10043	青岛	聊城	不详	不详	山东省	聊城市
10046	深圳	汕尾	广东	汕尾	广东省	汕尾市
1005	新乡	新乡	河南	新乡	河南省	新乡市

图 7.地理位置差异样例

3.2 成交时间特征

按日期统计训练集中每天借贷的成交量，正负样本分别统计，得到如下的曲线图 8，横坐标是日期（20131101 至 20141109），纵坐标是每天的借贷量。蓝色曲线是违约的样本每天的数量（为了对比明显，将数量乘上了 2），绿色曲线对应不违约的样本。

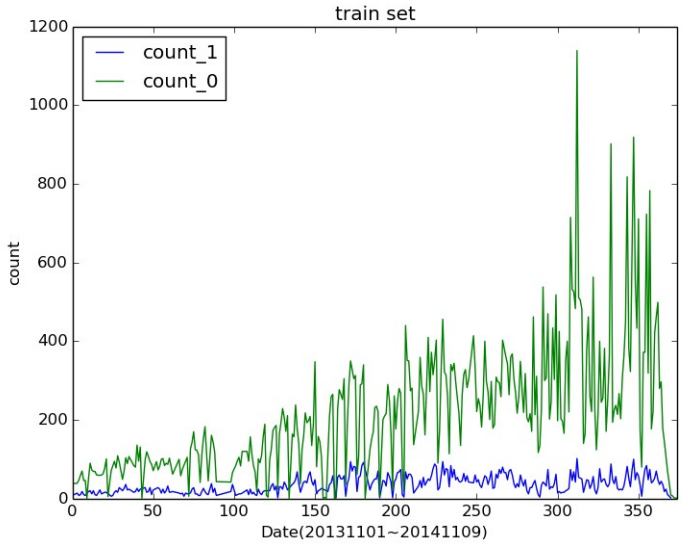

图 8.每日借贷量统计

可以发现拍拍贷的业务量总体是在增长的，而违约数量一开始也是缓慢增长，后面基本保持不变，总体上违约率是平稳甚至下降的。在横坐标 300～350 对应的日期区间，出现了一些借贷量非常大的时间节点，这些可能隐藏着某些信息，我们尚未挖掘出来。

考虑到违约率跟时间线有关，我们将成交时间的字段 Listinginfo 做了几种处理，一种 是直接将其当做连续值特征，也就是上图对应的横坐标，另一种是离散化处理，每 10 天作 为一个区间，也就是将日期0~10离散化为1，日期11~20离散化为2...

3.3 类别特征的处理

除了上面提到的对某些类别特征进行特殊处理外，其他类别特征都做独热编码。

3.4 组合特征

Xgboost 的训练完成后可以输出特征的重要性，我们发现第三方数据特征“ThirdParty_Info_Period_XX”的feature score比较大（见图3），即判别性比较高， 于是用这部分特征构建了组合特征：将特征两两相除得到7000+个特征，然后使用xgboost 对这 7000 多个特征单独训练模型，训练完成后得到特征重要性的排序，取其中top500个特 征线下cv能达到0.73+的AUC值。将这500个特征添加到原始特征体系中，线下cv的AUC 值从0.777提高到 0.7833。另外，也组合了乘法特征（取对数）：log(x*y)，刷选出其中 的270 多维，加入到原始特征体系中，单模型cv又提高到了0.785左右。

3.5 UpadteInfo 表特征

根据提供的修改信息表，我们从中抽取了用户的修改信息特征，比如：修改信息次数，修改信息时间到成交时间的跨度，每种信息的修改次数等等特征。

3.6 LogInfo 表特征

类似地，我们从登录信息表里提取了用户的登录信息特征，比如登录天数，平均登录间隔以及每种操作代码的次数等。

3.7 排序特征

对原始特征中190维数值型特征按数值从小到大进行排序，得到190维排序特征。排序 特征对异常数据有更强的鲁棒性，使得模型更加稳定，降低过拟合的风险。

四、特征选择

在特征工程部分，我们构建了一系列位置信息相关的特征、组合特征、成交时间特征、排序特征、类别稀疏特征、updateinfo 和 loginfo 相关的特征等，所有特征加起来将近 1500 维，这么多维特征一方面可能会导致维数灾难，另一方面很容易导致过拟合，需要做降维处 理，降维方法常用的有如 PCA，t-SNE 等，这类方法的计算复杂度比较高。并且根据以往经 验，在数据挖掘类的比赛中，PCA或t-SNE效果往往不好。

除了采用降维算法之外，也可以通过特征选择来降低特征维度。特征选择的方法很多：最大信息系数（MIC）、皮尔森相关系数（衡量变量间的线性相关性）、正则化方法（L1，L2）、基于模型的特征排序方法。比较高效的是最后一种，即基于学习模型的特征排序方法，这种方法有一个好处：模型学习的过程和特征选择的过程是同时进行的，因此我们采用这种 方法，基于xgboost 来做特征选择，xgboost 模型训练完成后可以输出特征的重要性（见 3.3 节图），据此我们可以保留Top N个特征，从而达到特征选择的目的。

五、类别不平衡的处理

赛题数据的类别比例接近 13:1，我们采用了两种解决类别不平衡问题的方法，一是在训练模型时设置类别权重，即代价敏感学习，二是采用过采样方法，本节主要介绍我们所采用的过采样方法。

SMOTE算法在合成小类样本时简便快捷，不容易造成过拟合。但SMOTE采样后虽然扩大 了小类的泛化空间，会同时缩小大类的泛化空间，以至于会降低对未知大类样本的预测准确 率，即存在一定盲目性。我们可以通过一个例子从邻域粗糙集的角度来分析其采样不当造成 新样本影响大类泛化空间的原因。

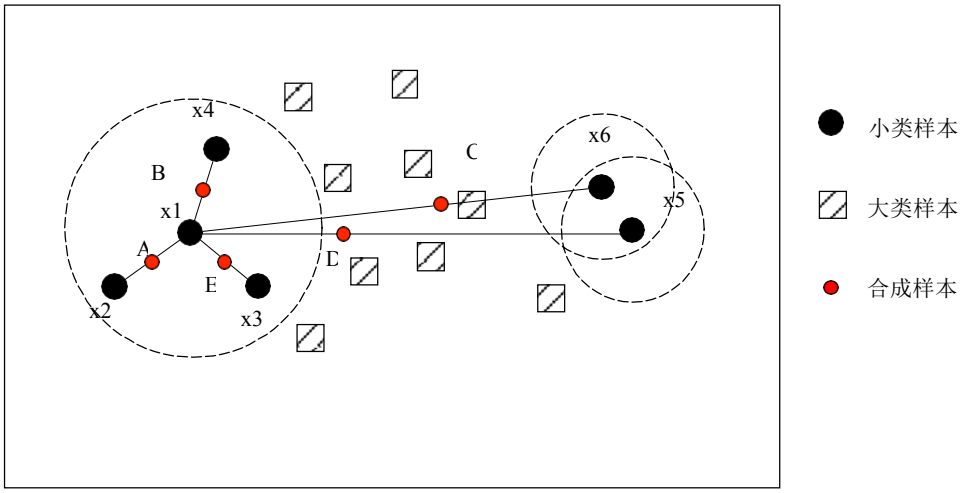

图9　SMOTE采样不当对大类泛化空间的影响

首先寻找小类样本x_1的同类K近邻（K=5），分别为$\{x_2 \sim x_6\}$，其中x_1与x_5，x_6都为小类正域样本，且x_5，x_6距离x_1较远，若使用这些点作为近邻采样，则新生成的样本点 C,D 与多类样本混叠在一起，影响大类样本的正常分类。

在比赛中我们采用了一种新的过采样方法，利用样本集内部的分布特性，然后通过样本邻域内样本分布来确定样本性质，对于那些邻域范围内既还有大类样本又含有小类样本的小类样本，即边界域样本，需要进行过采样。最后针对合成的新样本，若其对大类正域无影响，即该合成样本不包含在任意正域大类样本的邻域内，保留该样本，否则进行二次采样。采用这种方法后，线下cv有千分位的提升。

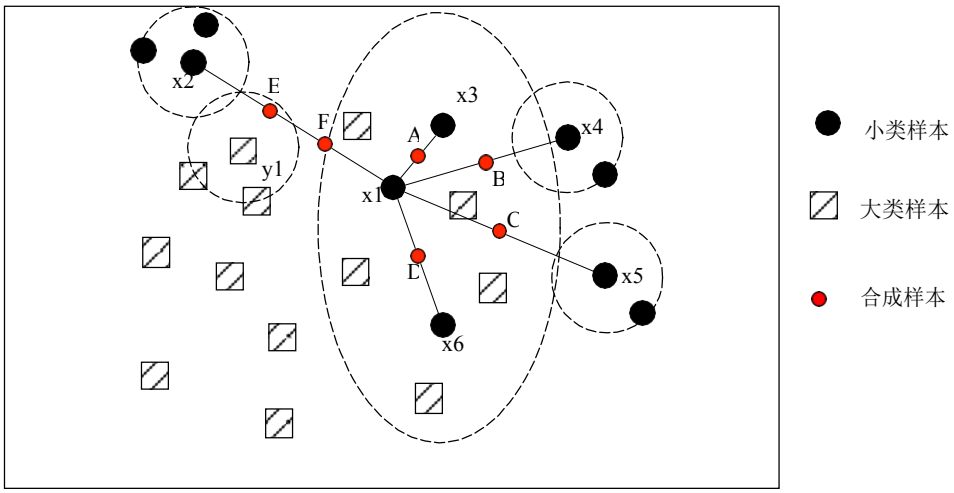

图 10边界采样原理图

图 10 中，椭圆区域内的样本都属于邻域粗糙集的边界域，而椭圆区域外的样本则都属于决策正域。当我们需要对边界域小类样本x_1进行采样时，首先找到它的同类K-近邻(k=5)，即图中的$\{x_2 \sim x_6\}$，若选择了$\{x_3 \sim x_6\}$其中的一个来合成新样本，可得到 A,B,C,D 四个合成样本其中的一个。可以看出，无论是哪个样本，都不会对决策正域内的大类样本产生影响，影响的只是边界域内的大类的泛化空间。当选择了，x_2若合成样本为E，则可以看出$E \in \delta(y)_1$，则在预测大类样本的时候，会使样本 y 的规则覆盖范围变小，影响对未知大类 样本的预测，导致错分。此时则需要进行二次采样，在x_1和x_2之间重新合成新样本，直到 能够合成出样本 F，即$F \notin \delta(y_1)$（重采样次数设置为5）。

六、模型设计与分析

复赛训练数据共有8万，测试数据1万。为了模拟线上提交，我们从8万训练数据中随机划分1万数据作为验证集，用于模拟线上提交，剩下的7万作为训练集，供模型调参使用。在7万训练集上采用7折交叉验证进行模型调参。得到多个单模型结果后，再在1万验证集 上做进一步的模型融合。以下将分别介绍。

6.1 逻辑回归

逻辑回归（logistic regression，以下简称 LR）可以说是互联网领域应用最广泛的分 类算法（没有之一），广告投放系统和推荐系统中的CTR预估模型的主干算法基本都是LR。 尽管LR在比赛中并非大杀器，但从实用性角度出发，还是值得去探索的。

对于原始特征和构建的 1439 维特征进行添加和形式的变换。动态地根据模型的 cross validation 反馈来处理特征，贯穿整个建模过程，对于 xgboost 模型训练完成后输出的重 要性特征，进行单因素分析、数值特征的离散化，对于连续性变量在分析过程中常常需要进 行离散变成等级特征然后进行独热编码，其中使用 OCDD 离散化方法进行离散。构建并加入上述 4.4 中描述的组合特征，由于线性模型对于非线性关系缺乏准确刻画，特征组合正好可

以加入非线性表达，增强模型的表达能力。 模型的参数过多时，很容易遇到过拟合的问题。

需要控制模型的复杂度，典型的做法在
优化目标中加入正则项，通过惩罚过大的参数来防止过拟合，L1正则化倾向于使参数变为0，可以进行特征选择，对高纬度数数据有不错的效果，故我们选取L1正则。

最后在1700维特征上训练模型，单模型验证集上我们得到的AUC值为0.772左右。

6.2 XGBOOST

xgboost 是 boosted tree 的一种实现，效率和精度都很高，在各类数据挖掘竞赛中被 广泛使用。由于xgboost支持分布式运行，很多互联网企业也在实际业务中采用了xgboost， 实用性很高，比如阿里巴巴的ODPS平台部署了xgboost，腾讯在微信购物里用xgboost做 了CTR预估。

基于前面构建出的特征，加上原始特征共有 1439 维特征，在这 1439 维特征上我们训练了 xgboost，单模型线下 cross validation 的 auc 值为 0.7833 左右（在下文中，称此模型 为 xgb_7833）。

受bagging思想的启发，我们对单模型xgb_7833做了进一步的改进，xgb_7833确定了 一组不错的参数，我们让这些参数在一定的小范围内随机波动，同时对特征进行随机抽样， 训练多个xgb子模型进行bagging。例如，xgb_7833的subsample参数取值为0.75,而子xgb 模型的subsample参数则在(0.7,0.8)之间随机取值；xgb_7833所用到的特征为1439维， 而子 xgb 模型则随机抽样部分特征进行训练。这种方法在参数和特征上都引入了多样性（差 异性），使得最后bagging的效果有很大的提升，线下cv达到了0.787的auc值。该模型 框图如下所示：

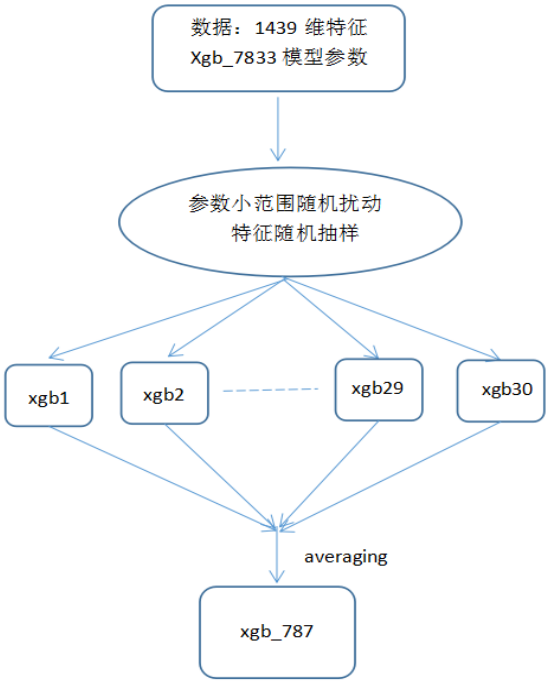

图11. Xgb模型bagging

6.3 Large-Scale SVM 的尝试

支持向量机（SVM）在处理大规模数据问题时存在训练时间过长和内存空间需求过大的 问题，当数据量过万时，普通 PC 跑起来很吃力。本次赛题提供的训练数据有 8 万，对 SVM 来说可以算是小的大规模数据了。SVM 解决 large-scale 数据问题，一般采用工作集方法、 训练集分解方法、增量学习方法等，在这次比赛中我们采用了训练集分解的方法（参考了论 文 Ensemble SVM的思想），以下是所采用的方法的简要框图：

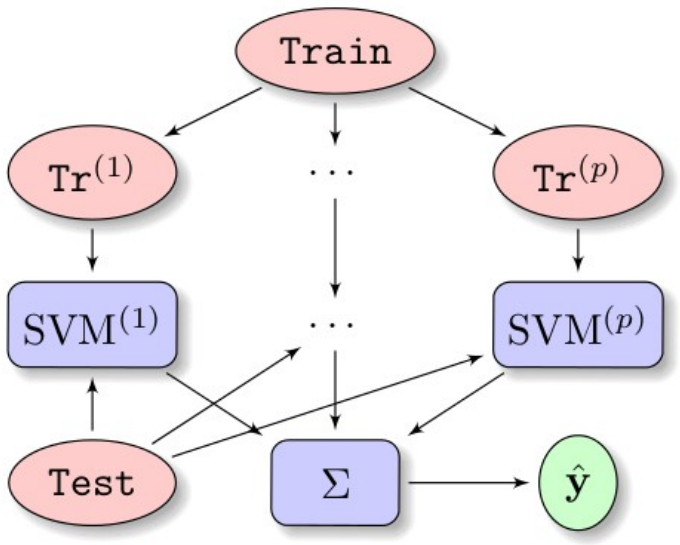

图 12. Large-ScaleSVM

我们训练了svm^{1}、svm^{2}..... svm^{30}共30个svm，每个svm所用到的训练数据只是原始数据 的一个子集，子集产生的方式是 Bootstrap抽样。这种方法看似增加了计算复杂度，实际上是减小的，假设原始训练数据大小是 n，则在原始数据上训练的复杂度是 $O(n^2)$，将数据集n分成p份，则每份数据量是 $\frac{n}{p}$，每一份训练一个子svm，复杂度是$O\left(\left(\frac{n}{p}\right)^2\right)$，全加起来是$\sum_{i=1}^{p} O\left(\left(\frac{n}{p}\right)^2\right) = O\left(n^2/p\right)$，复杂度比在原始数据上训练减小了 p倍。

得到 30 个子 svm 的预测结果后，我们对它们进行简单的取平均，这种方法在验证集上得到的AUC是0.77左右。这种方法在保证了准确率的基础上，极大地减少了训练时间。

6.4 多模型 blending

在 Kaggle 等高水平的数据挖掘大赛中，诸多冠军队伍都会使用多模型融合技术，比如 blending ensemble 和 stack ensemble 技术，我们知道单个模型的结果不够理想，如果想 得到更好的结果，需要把很多单个模型的结果融合在一起，这就是一种 ensemble 技术。我 们在比赛中选择 blending ensemble 的 ensemble 方式，主要融合效果较好 4 个树模型：

xgboost > GDBT > ExtraTrees > RandomForest。

Blending过程主要根据CV过程将train数据集拆分成子训练集train_train和子验证 数据集train_valid,通过每次对 train_train 进行训练分别对 train_valid 和 valid 数据 集进行预测train_valid_pred和val_pred结果，将每个模型CV产生的train_valid_pred 拼接成 train_pred，对 val_pred 取平均生成 val_pred_mean，上层模型可以选用 LogisticRegression对每个模型产生的train_pred与train_y进行训练，对val_pred_mean 进行预测生成答案。

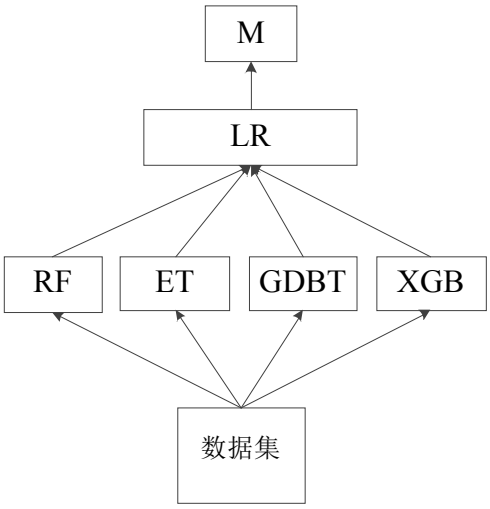

图13. 多模型blendin ensemble

6.5 模型融合

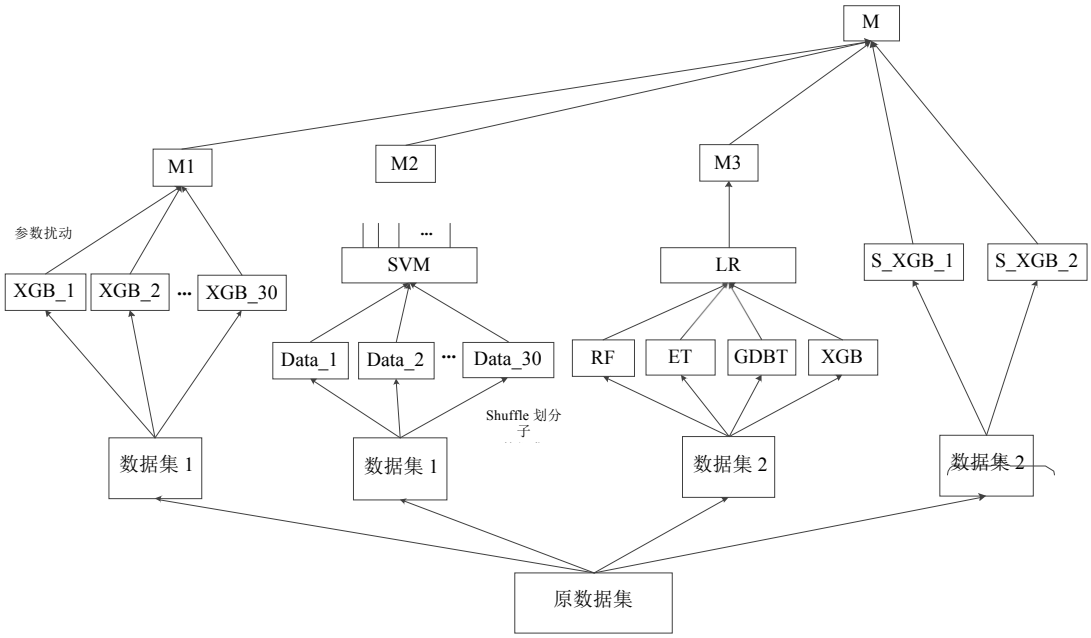

图 14.模型融合总框架

上图所示，我们的框架主要实现2层的多模型融合,基于4个方案结果的融合：

- 方案1：bagging xboost，通过参数扰动生成不同30个xgb模型，对同一数据集进行 训练产生结果，采用均值融合产生M1结果
- 方案2：Large-ScaleSVM，通过shuffle将数据集划分为30子数据集，根据复杂度分 析，这个方式的时间复杂度进一步降低，对30个svm结果输出取平均产生M2结果
- 方案 3：多模型 Blending，由于不同模型具备极大差异性，通过不同模型对同一数据集 进行训练产生结果作为元特征继续训练，上层采用LR对输出的特征进行权重训练学习 克服手工选择权重的问题，产生M3模型
- 方案4：单模型，根据线下CV验证，调试出2个不同版本的最优单模型，S_XGB_1, S_XGB_1 分别为陈天奇版本的 xgboost 和 graphlab 版本的模型，生成不同的单模型结果。 模型融合的关键是在于模型差异性。框架中体现出模型差异性主要在：不同模型，相同

模型不同参数，训练数据集不同，这3个部分。因此每次在ensemble之前需要参考下各个 模型的相关性，可以采用cos相似或者person系数评价。

最上层的融合方式可以采用均值融合，rank 均值融合等简单方式，最终我们选用的是线下效果最好的1/rank加权融合（按score降序）。

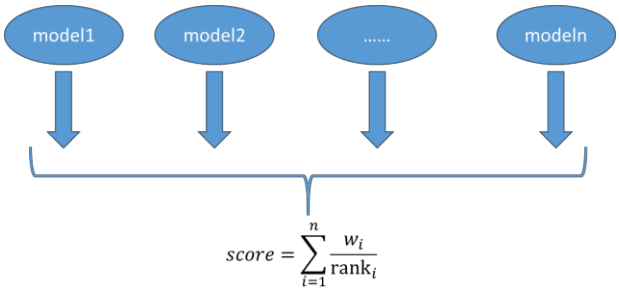

图 15.上层融合方案