Kaggle CTR 预估竞赛统计文档

（新浪门户广告算法组）

文档名称	版本号	作者	修改&评审人
Kaggle CTR	1.0	周永	

1. 解决思路

我们主要从以下 4 个主要思路来解决这个问题，即挖掘特征，尝试 LR、FTRL、SVM、NB、LibFM 等不同的预估模型，后续使用 Random Forest, GBDT 等集成方法做模型 Ensemble，最后通过调参、交叉验证等工作优化指标。下面分别阐述：

1.1 特征挖掘

我们提取了 6 大类主要的特征：

(1). **Basic Sparse Features**

基本的稀疏特征，采用 one-hot encoding 方法对连续型特征进行编码，得到稀疏特征，具体的编码过程见 2.1 第 1 版提交 结果。

(2). **Click-Through Rate Features**

统计每一个特征平滑之后对应的历史 pseudo-ctr 值，作为 ctr 特征。

(3). **Discretized-CTR Features**

为提高不同 feature 的 pseudo-ctr 值的区分度，将(2)生成的历史 ctr 特征，进行**特征离散化**。

(4). **Numerial Features + Normalization**

在原始数据中，通过分析提取**数值类特征**，包括 C21~C14，C1 匿名类特征、广告位置 banner_pos, 广告相对位置 relative_pos 等，对数值类特征做了**特征归一化**.

(5). **组合特征：Two-Combined Features**

数据中明显有 app，site，device 三类特征，通过**两两特征组合**，得到 app-site, app-device, site-device **3 大类组合特征**和基于广告位的两两特征组合。

(6). **组合特征：Multi-Combined Features**

在两两组合特征的基础上，获取多个特征的组合。

1.2 预估模型

在原始数据和挖掘的特征数据基础上，我们尝试了很多 regression, classification, libfm 分解机器等方法，具体的预估模型尝试结果见第 2 部分的提交结果。

1.3 集成方法

通常多模型预估结果与单一模型的预估结果相比，其 variance 和稳定性都要好，同时也借鉴了 Kaggle 历届数据挖掘竞赛的相关经验，我们尝试了

RandomForest, GBDT, Boosting 等集成方法，通过调参获取每一个模型的最优结果。

1.4 交叉验证与特征选择

通过划分验证集的方式，根据模型、参数在验证集上的表现，作为进一步调参的依据。

2. 提交结果

2.1 第 1 版提交

<table>
<tr><td>版本号</td><td>1</td><td>提交日期</td><td>2014-11-27</td></tr>
<tr><td colspan="4">技术细节</td></tr>
<tr><td>特征方面</td><td colspan="3">(1). Basic Sparse Features:
在原有 23 个 categorical feature 中，选择 18 个 feature，进行 one-hot 编码，编码后的 sparse binary features 数量为 26238 维。18 个特征情况：
⑴ 匿名离散特征：C1，C14 ~ C21；
⑵ 广告位特征：banner_pos;
⑶ 站点特征：site_id, site_domain, site_category;
⑷ app 特征：app_id, app_domain, app_category;
⑸ device 特征：device_type, device_conn_type.
(2). 未使用特征：
⑴ adid 特征：由于训练集中每个 ad 仅对应一条记录，one-hot 维度太高，且对结果无意义；
⑵ time 特征：特征值类型 yymmddhh, training data 日期与 testing data 日期无重合，暂且不用，下一版会提取 hh 字段，然后再 one-hot 编码；
⑶ device_id, device_ip, device_model: 每个特征取值有数百万，三个 device 特征 one-hot 编码后，超过 1000 万维，在 spark 上跑 LR 时，耗时太长且容易跑崩，暂且没用。后续会用统计平滑方法得到这三个特征的历史 CTR 值；
(3). 其它：
特征编码中涉及生成<feature,index>映射表，生成编码后的训练数据和测试数据等。</td></tr>
<tr><td>模型方面</td><td colspan="3">Logistic Regression：利用 Spark 上的 LR 训练模型，生成提</td></tr>
</table>

	交结果。 （在 10m 特征下，没跑成功，可能是对 spark RDD 使用不够深入，亦有可能特征维度太高，需要特殊处理一下，比如降维 or 特征 hashing 等。）				
评估结果					当时排名
提交	LogLoss	**0.4322163**			**312**
离线	LogLoss	暂时无	AUC	暂时无	

2.2 第 2 版提交

版本号	2	提交日期	2014-12-03
技术细节			
特征方面	(1). **Basic Sparse Features:** 在第 1 版特征 one-hot 编码基础上，添加 **hour 和 device_model** one-hot 编码特征，新增 24+8303=8327 维特征； (2). **Click-Through Rate Features**: For each feature,计算平滑之后的历史 CTR（pseudo-ctr），总共得到 22 维特征； 目前为止总的特征数：**34587 维**		
模型方面	**Logistic Regression**：利用 Spark 上的 LR 训练模型，生成提交结果。		

评估结果					当时排名
提交	LogLoss	**0.4316984**			**385**
离线	LogLoss	0.4401297	AUC	0.695723	

2.3 第 3 版提交

版本号	3	提交日期	2014-12-06
技术细节			
特征方面	(1). **Basic Sparse Features (34565 维):** 除 device_id 和 device_ip 非常稀疏的特征外，其它全部特征用于生成 one-hot 编码特征； (2). **Click-Through Rate Features (22 维):** For each feature,计算平滑之后的历史 CTR（pseudo-ctr），		

	总共得到 22 维特征； (3). **Discretized-CTR Features (872 维)：** 为提高不同 feature 的 pseudo-ctr 值的区分度， 将(2)生成的历史 ctr 特征，进行**特征离散化**，得到了 872 维 ctr 离散化后的特征。 目前为止总的特征数：**35459 维**。
模型方面	**Logistic Regression**：利用 Spark 上的 LR 训练模型，生成提交结果。

评估结果					当时排名
提交	LogLoss	**0.4253557**	提升	0.0063427	**416**
离线	LogLoss	0.4255774	AUC	0.714115	

2.4 第 4 版提交（高翔）

2.5 第 5 版提交（周永）

版本号	5	提交日期	2014-12-09
技术细节			
特征方面	(1). **Basic Sparse Features (34565 维):** 除 device_id 和 device_ip 非常稀疏的特征外，其它全部特征用于生成 **one-hot 编码**特征； (2). **Click-Through Rate Features (22 维):** For each feature,计算**平滑**之后的历史 CTR（pseudo-ctr），总共得到 22 维特征； (3). **Discretized-CTR Features (872 维)：** 为提高不同 feature 的 pseudo-ctr 值的区分度， 将(2)生成的历史 ctr 特征，进行**特征离散化**，得到了 872 维 ctr 离散化后的特征。 (4) **Numerial Features + Normalization（13 维）** 在原始数据中，通过分析提取**数值类特征**，包括 C21~C14，C1 匿名类特征、广告位置 banner_pos, 广告相对位置 relative_pos 等，对数值类特征做了**特征归一化**，得到了 13 维数值特征。 目前为止总共的特征数：**35472 维**。		
模型方面	**Logistic Regression**：利用 Spark 上的 LR 训练模型，生成提交结果。		

评估结果					当时排名
提交	LogLoss	**0.4243063**	提升	0.0010494	**480**
离线	LogLoss	0.4256732	AUC	0.7108962	
结论	添加的数值类特征，在原有参数不变的情况下，指标从 0.425 提升至 0.424, 效果不是很明显。分析一下，原始数据都是一些 category 特征，这里只是把一些用整数表示特征的离散值当做连续值使用，然后做了归一化，效果提升不明显在意料之中。但如果数据中有连续值特征，抽取数值类特征在 feature engineering 中是不可缺少的一步。				

2.6 第 6 版提交

版本号	6	提交日期	2014-12-11
技术细节			
特征方面	(1). **Basic Sparse Features (34565 维):** 除 device_id 和 device_ip 非常稀疏的特征外，其它全部特征用于生成 **one-hot 编码**特征； (2). **Click-Through Rate Features (22 维):** For each feature,计算**平滑**之后的历史 CTR（pseudo-ctr），总共得到 22 维特征； (3). **Discretized-CTR Features (872 维)：** 为提高不同 feature 的 pseudo-ctr 值的区分度， 将(2)生成的历史 ctr 特征，进行**特征离散化**，得到了 872 维 ctr 离散化后的特征。 (4) **Numerial Features + Normalization (13 维)** 在原始数据中，通过分析提取**数值类特征**，包括 C21~C14，C1 匿名类特征、广告位置 banner_pos, 广告相对位置 relative_pos 等，对数值类特征做了**特征归一化**，得到了 13 维数值特征。 目前为止总共的特征数：**35472 维**。		
模型方面	**Logistic Regression**：利用 Spark 上的 LR 训练模型，生成提交结果。 **调参：调整正则化项系数，regParam=0.08 (之前为 0.1)**		
评估结果			当时排名

提交	LogLoss	**0.4226858**	提升	0.0016205	**655**
离线	LogLoss	0.42300388	AUC	0.7194154	
结论	这一版主要是调参，特征总维度是 35472，而训练数据是 4000 万条记录，使用像 lr 这种线性模型作为预估模型，过拟合的可能性比较小。因此这里降低正则化项的比重（减小正则化因子系数的值），发现预估效果有提升，下一步继续调正则化因子系数，观察结果。				

2.7 第 7 版提交

版本号	7	提交日期	2014-12-15
技术细节			
特征方面	(1). **Basic Sparse Features (34565 维，20 列):** 除 device_id 和 device_ip 非常稀疏的特征外，其它全部特征用于生成 **one-hot 编码**特征； (2). **Click-Through Rate Features (22 维，22 列):** For each feature,计算**平滑**之后的历史 CTR（pseudo-ctr），总共得到 22 维特征； (3). **Discretized-CTR Features (872 维，22 列)：** 为提高不同 feature 的 pseudo-ctr 值的区分度， 将(2)生成的历史 ctr 特征，进行**特征离散化**，得到了 872 维 ctr 离散化后的特征。 (4) **Numerial Features + Normalization (13 维，13 列)** 在原始数据中，通过分析提取**数值类特征**，包括 C21~C14，C1 匿名类特征、广告位置 banner_pos, 广告相对位置 relative_pos 等，对数值类特征做了**特征归一化**，得到了 13 维数值特征。 目前为止总共的特征数：**35472 维**。		
模型方面	**Logistic Regression**：利用 Spark 上的 LR 训练模型，生成提交结果。 **调参：调整正则化项系数，regParam=0.05 (之前为 0.08)** **训练集：验证集 = 9.9:0.1**		
评估结果			当时排名
提交	LogLoss	**0.4194096**	提升	0.0032762	**629**

离线	LogLoss	0.41572516	AUC	0.7370057	
	平均 ctr	0.175208			
结论	再次降低正则化因子系数（即 regParam），从 0.08 降到 0.05，发现，预估效果又有了比较大的提升，验证了之前的想法，就是说，对于线性模型，当样本数（行数）远大于特征数（列数）时，过拟合的可能性就越小，对应的正则化项的作用就越小。				

2.8 第 8 版提交

版本号	8	提交日期	2014-12-16
技术细节			
特征方面	(1). **Basic Sparse Features (34565 维，20 列):** 除 device_id 和 device_ip 非常稀疏的特征外，其它全部特征用于生成 **one-hot 编码**特征； (2). **Click-Through Rate Features (22 维，22 列):** For each feature,计算**平滑**之后的历史 CTR（pseudo-ctr），总共得到 22 维特征； (3). **Discretized-CTR Features (872 维，22 列)：** 为提高不同 feature 的 pseudo-ctr 值的区分度， 将(2)生成的历史 ctr 特征，进行**特征离散化**，得到了 872 维 ctr 离散化后的特征。 (4) **Numerial Features + Normalization（13 维，13 列）** 在原始数据中，通过分析提取**数值类特征**，包括 C21~C14，C1 匿名类特征、广告位置 banner_pos, 广告相对位置 relative_pos 等，对数值类特征做了**特征归一化**，得到了 13 维数值特征。 目前为止总共的特征数：**35472 维**。		
模型方面	**Logistic Regression**：利用 Spark 上的 LR 训练模型，生成提交结果。 **调参：调整正则化项系数，regParam=0.05 (之前为 0.08)** **训练集：验证集 = 9:1**		
评估结果			当时排名
提交	LogLoss	**0.4194104**	提升	**没有提升**	**649**
离线	LogLoss	0.41572516	AUC	0.7370057	

	平均 ctr	0.175223			
结论	这里在原有基础上，调整训练集与验证集的比例，参数设置为 9:1 时，验证集的样本数与测试集的样本数基本一致，分别观察离线和在线的 logloss 值，发现离线评估结果和线上结果基本相同，这说明验证集与测试集的数据分布是一致的，从训练集中抽 10%的数据用于离线预测是合理的.				

2.9 第 9 版提交

版本号	9	提交日期	2014-12-17
技术细节			
特征方面	(1). **Basic Sparse Features (34565 维，20 列):** 除 device_id 和 device_ip 非常稀疏的特征外，其它全部特征用于生成 **one-hot 编码**特征； (2). **Click-Through Rate Features (22 维，22 列):** For each feature,计算**平滑**之后的历史 CTR（pseudo-ctr），总共得到 22 维特征； (3). **Discretized-CTR Features (872 维，22 列)：** 为提高不同 feature 的 pseudo-ctr 值的区分度， 将(2)生成的历史 ctr 特征，进行**特征离散化**，得到了 872 维 ctr 离散化后的特征。 (4) **Numerial Features + Normalization（13 维，13 列）** 在原始数据中，通过分析提取**数值类特征**，包括 C21~C14，C1 匿名类特征、广告位置 banner_pos, 广告相对位置 relative_pos 等，对数值类特征做了**特征归一化**，得到了 13 维数值特征。 目前为止总共的特征数：**35472 维**。		
模型方面	**Logistic Regression**：利用 Spark 上的 LR 训练模型，生成提交结果。 **调参：调整正则化项系数，regParam=0.03 (之前为 0.05)** **训练集：验证集 = 9:1**		

评估结果					当时排名
提交	LogLoss	**0.4161412**	提升	0.0032684	**621**
离线	LogLoss	0.40588654	AUC	0.7584057	

	平均 ctr	0.175065			
结论	原有基础上，再次降低正则化项的比重，参数设置为 0.03. 效果又提升一些。				

2.10 第 10 版提交

版本号	10/27		提交日期	2014-12-17
技术细节				
特征方面	(1). **Basic Sparse Features (34565 维，20 列):** (2). **Click-Through Rate Features (22 维，22 列):** (3). **Discretized-CTR Features (872 维，22 列)：** (4) **Numerial Features + Normalization (13 维，13 列)** 目前为止总共的特征数：**35472 维**。			
模型方面	**Logistic Regression**：利用 Spark 上的 LR 训练模型，生成提交结果。 **调参：调整正则化项系数，regParam=0.01 (之前为 0.03)** **训练集：验证集 = 9:1**			
评估结果				当时排名
提交	LogLoss	**0.4111210**	提升	0.0050202	**624**
离线	LogLoss	0.38018302	AUC	0.80246755	
	平均 ctr	0.175832			
结论	原有基础上，再次降低正则化项的比重，参数设置为 0.01. 效果又提升一些，提交版本的平均 ctr 有所上升，这说明了平均 ctr 的大小与最后提交的 logloss 结果无必然关系。 ---------- 同样尝试了 regParam = 0.02, 9:1 的结果：记录如下： 提交 LogLoss：0.4138844；离线 logloss：0.775300625；离线 AUC：0.39720846；平均 ctr：0.175317。(第 26 次提交)			

【20141217】Spark LR 调参

RegParam: 正则化项系数；

Train:CV：训练集与验证集的比例；

regParam	Train:CV	离线 log	Auc	平均 ctr	提交 log
0.02	9:1	0.39720846	0.7753006	0.175317	0.4138844
0.01	9:1	0.38018302	0.80246755	0.175832	**0.4111210**

0.0005	9:1	0.31300838	0.86732296	0.172219	0.4298685
当 regParam=0.0005 时，提交的 logloss 变大了，过拟合？					
20141218 结果					
0.005	9:1	0.36180885	0.82567285	0.175805	**0.4103477**
0.002 (30)	9:1				0.4135403
当 regParam=0.002 时，同样出现了提交的 logloss 值变大的情况，看来是过拟合了？					
0.004 (31)	9:1			0.175623	0.4107100
0.006 (33)	9:1	0.36666868	0.81991194	0.175845	
0.005 **(32)**	9.99:0.01	0.36118534	0.83098283	0.173389	**0.4079313**
32 次提交是用了 combined 特征，5 大类特征~， 32 之前和 33 都是 4 大类 feature。					

2.11 第 11 版提交

版本号	11/29	提交日期	2014-12-18
技术细节			
特征方面	(1). **Basic Sparse Features (34565 维，20 列):** (2). **Click-Through Rate Features (22 维，22 列):** (3). **Discretized-CTR Features (872 维，22 列)：** (4) **Numerial Features + Normalization (13 维，13 列)** 目前为止总共的特征数：**35472 维。**		
模型方面	**Logistic Regression**：利用 Spark 上的 LR 训练模型，生成提交结果。 **调参：调整正则化项系数，regParam=0.005 (之前为 0.01)** **训练集：验证集 = 9:1**		

评估结果					当时排名
提交	LogLoss	**0.4103477**	提升	0.0007733	**628**
离线	LogLoss	0.36180885	AUC	0.82567285	
	平均 ctr	0.175805			

结论	原有基础上，再次降低正则化项的比重，参数设置为 0.01. 效果又提升一些，提交版本的平均 ctr 有所上升，这说明了平均 ctr 的大小与最后提交的 logloss 无必然关系。 ---------- 同样尝试了 regParam = 0.02, 9:1 的结果：记录如下： 提交 LogLoss：0.4138844；离线 logloss：0.775300625；离

	线 AUC：0.39720846；平均 ctr：0.175317。

2.12 第 12 版提交

版本号	12/32	提交日期	2014-12-19
技术细节			
特征方面	(1). **Basic Sparse Features (34565 维，20 列):** (2). **Click-Through Rate Features (22 维，22 列):** (3). **Discretized-CTR Features (872 维，22 列)：** (4) **Numerial Features + Normalization (13 维，13 列)** (5) **Two-Combined Features (110572 维，21 列)** 数据中明显有 app，site，device 三类特征，通过**两两特征组合**，得到 app-site, app-device, site-device **3 大类组合特征。**具体地： app: {app_id, app_domain, app_category}; site: {site_id, site_domain, site_category}; device: {device_type, device_conn_type} 总共**获得 21 个组合特征** ,onehot 编码后 ,转换得到 **110572 维 sparse binary features.** 下一步，原始特征中还有 hour, banner_pos, C1, C14-C21 匿名特征等未使用，下一步会抽取**广告位 banner_pos 与 app,site,device 三类特征之间的组合特征。** 后续，会分析两两特征之间和多个特征之间的**相关性**，抽取相关性较大的多特征之间的组合特征。得到**相关性特征**和**多维组合特征。** 目前为止总共的(稀疏)特征数：**146 044 维。**		
模型方面	**Logistic Regression**：利用 Spark 上的 LR 训练模型，生成提交结果。 **调参：调整正则化项系数，regParam=0.005** **训练集：验证集 = 9:1**		

评估结果					当时排名
提交	LogLoss	**0.4079313**	提升	0.0024164	**631**
离线	LogLoss	0.36118534	AUC	0.83098283	
	平均 ctr	0.173389			

结论	与上一版相比，在原有参数不变的情况下，添加 two-combined 特征，提交的 logloss 值变小，说明该部分组合特征对于 click 是有效果的。

版次	regParam	Train:CV	离线 log	Auc	平均 ctr	提交 log
以下都是包含 combined 特征的 5 大类特征，						
32	0.005	9.99:0.01	0.36118534	0.83098283	0.173389	0.4079313
34	0.006	9.99:0.01	0.36588799	0.82566676	0.173557	**0.4077505**
35	0.01	9.99:0.01			0.173625	0.4080480
【20141229】34 与 35 相比，正则化项提升至 0.01 后，指标变得更差。说明 regParam 在 0.006~0.01 之间。						

2.13 第 13 版提交

版本号	13/36	提交日期	2014-12-29
技术细节			
特征方面	(1). **Basic Sparse Features (34565 维，20 列):** (2). **Click-Through Rate Features (22 维，22 列):** (3). **Discretized-CTR Features (872 维，22 列)：** (4). **Numerial Features + Normalization（13 维，13 列）** (5). **Two-Combined Features（110572 维，21 列）** (6). **Banner_pos-based two-combined features （31128 维，17 列）** 从原始数据中抽取与广告位 banner_pos 相关的组合特征。数据中的特征类型： app: {app_id, app_domain, app_category}; site: {site_id, site_domain, site_category}; device: {device_type, device_conn_type}; 匿名类（anony）：{C1, C14, C15, …, C21} 抽取的广告位组合特征有：banner_pos-app, banner_pos-site, banner_pos-device, banner_pos-anony. 基于广告位的两两组合特征列数: 3+3+2+9=17 个。转换得到 31128 维 sparse binary features.		

	后续，会分析两两特征之间和多个特征之间的**相关性**，抽取相关性较大的多特征之间的组合特征。得到**相关性特征**和**多维组合特征**。 目前为止总共的(稀疏)特征数：**177 172 维。**
模型方面	**Logistic Regression**：利用 Spark 上的 LR 训练模型，生成提交结果。 **调参：调整正则化项系数，regParam=0.01** **训练集：验证集 = 9.99:0.01**

评估结果					当时排名
提交	LogLoss	**0.4070954**	提升	0.0006551	**751**
离线	LogLoss	0.382434297	AUC	0.80070664	
	平均 ctr	0.172440			
结论	与上一版相比，在原有参数不变的情况下，添加 two-combined 特征，提交的 logloss 值变小，说明该部分组合特征对于 click 是有效果的。				

【20141229】Spark LR 调参

使用我们生成的 6 组特征进行试验

版次	regParam	Train:CV	离线 log	Auc	平均 ctr	提交 log
以下都是包含 combined 特征的 6 大类特征，						
37	0.006	9: 1	0.36501926	0.81890896	0.172189	**0.4068321**
提升：0.0002633						
38	0.008	9.99:0.01	0.37681751	0.80821919	0.172386	0.4068399
【20141229】34 与 35 相比，正则化项提升至 0.01 后，指标变得更差。说明 regParam 在 0.006~0.01 之间。						
39	0.005	9.99:0.01	0.36468953	0.82345866	0.172292	

2.14 第 14 版提交

版本号	14/40	提交日期	2015-01-10
技术细节			
特征方面	原始数据		
模型方面	**FTRL：调参 L2=0.5（之前是 1）**		

评估结果					当时排名
提交	LogLoss	**0.3956172**	提升	**0.0112149**	**458**

离线	LogLoss		AUC		
	平均 ctr	0.172956			
结论					

2.15 FTRL 预估模型调参系列：【20150112】

特征代号：

原始: 最初的原始特征

F2: onehot_pseudoctr

F3: onehot_pseudoctr_discretization

F4: onehot_pseudoctr_discretization_numerical

F5: onehot_pseudoctr_discretization_numerical_combined

特征	D(hash)	L1	L2	alpha	Mean_ctr	Logloss
原始	2 ** 20	1.0	0.5	0.1	0.172956	0.3956172
原始	2 ** 20	1.0	2.0	0.1	0.172339	0.3955296
原始	**2 ** 25**	1.0	1.0	0.1	0.172810	0.3948784
原始	2 ** 25	1.0	**2.0**	0.1	0.172401	0.3948551
原始	2 ** 25	**2.0**	**2.0**	0.1	0.170493	0.3947960
原始	2 ** 25	**1.0**	**4.0**	0.1	0.171660	0.3948192
原始	2 ** 25	**2.0**	**4.0**	0.1	0.169918	**0.3947821**
分析：L1=2 时要比 L1=1 时效果更好，这说明 L1 权重越大，解决稀疏性就越好，同时也说明了数据是稀疏的，因此可以将 L1=4 再查看结果....						
原始 51	2 ** 25	**4.0**	**4.0**	0.1		0.3949060
原始 52	2 ** 25	**8.0**	**4.0**	0.1		
分析：从 51 次提交来看，L1=4 时效果又不好了，可能过了，因此 L1=3 时，再试试，L1=3 时效果更高						
原始 53	2 ** 25	**3.0**	**4.0**	0.1		**0.3947748**
原始 54	2 ** 25	**2.5**	**4.0**	0.1	0.169239	0.3947758
原始 55	2 ** 25	**2.5**	**8.0**	0.1		0.3947748,
原始 56	2 ** 25	**3.5**	**4.0**	0.1		
分析：53 次对应的 L1=2.5,说明效果又提升，下面尝试 L1=3 即 53 的情况						
F2	2 ** 20	1.0	2.0	0.1	0.170918	**0.4025958**
F2	2 ** 20	**0.1**	**0.1**	**0.1**		0.4030509
F2	2 ** 20	**0.1**	**0.0**	**0.1**		0.4030612

F2(54)	2 ** 20	**1.0**	**4.0**	**0.1**		
F2						
F5	-----	1.0	1.0	0.1	0.170809	**0.5557508**
结果很糟糕..... 分析：之所以对原始数据的 L1 和 L2 增大，是因为特征数在百万级，为了防止过拟合；而使用我们的编码数据，特征维数为数万，如果还使用之前的 L1 和 L2 的惩罚力度，可能会导致欠拟合，也许这是导致 FTRL+F5 效果不好的原因。因此，调整 L1 和 L2 的系数，使之比重下降。						
F5	-----	1.0	2.0	0.1		

2.16 模型组合：RandomForest && LibFMC（20150113~）

（正在总结中...）

【失败尝试】GBRT 的失败尝试

使用 xgboost 开源库中的 gradient boosting regression tree(GBRT)算法，用我们生成的数据 one-hot + pseudo_ctr 特征，训练模型，统计结果：

序号	tree_depth	num_round	logloss	Mean_ctr
1	20	2	0.5246676	0.190957
2	10	10	0.5984752	
3	6	6	0.5551602	
4	50	6	0.71**	

说明序号 4：当 tree_depth=50，在 cv 阶段，logloss 值先变小后变大，说明模型可能过拟合。得到的结果是 0.71*，这说明模型过拟合了。因此需要减少树的 depth.