

如何解决ad click prediction system中， Imbalanced Data: Click events are sparse relative to impressions, requiring techniques to address imbalance的问题？

在广告点击预测（Ad Click Prediction）系统中，点击事件（正样本）通常只占曝光量的1%以下，这种极端的数据不平衡会导致模型偏向负样本，降低对正样本的识别能力。以下是针对该问题的系统性解决方案：

一、数据层面的处理

1. 重采样（Resampling）

过采样（Oversampling）：

SMOTE：合成少数类样本（适用于中等规模数据）。

复制正样本：简单但可能过拟合。

自适应合成采样（ADASYN）：根据样本密度动态生成数据。

欠采样（Undersampling）：

随机欠采样：随机丢弃部分负样本（可能丢失重要信息）。

NearMiss：保留与正样本最相关的负样本。

混合采样：结合过采样和欠采样。

2. 动态采样（Online Sampling）

流式数据采样：实时调整采样率（如负样本下采样至10%~20%比例）。

Batch内平衡：每个训练batch强制保持正负样本比例（如1:1）。

3. 数据增强（Data Augmentation）

特征空间变换：对正样本进行轻微扰动（如数值特征加噪声）。

GAN生成样本：使用生成对抗网络合成点击样本（适用于高维特征）。

4. 類別權重 (Class Weights): 在模型的損失函數中為不同類別分配不同的權重。通常給予少數類別更高的權重，使其錯誤分類的懲罰更大。這在許多模型中都可用，例如邏輯迴歸、支援向量機(SVM)、決策樹集成模型(XGBoost, LightGBM)。

- 優點： 直接影響模型優化過程，無需修改數據集。

- 實例：在 Scikit-learn 的 `LogisticRegression` 或 `SVC` 中設置 `class_weight='balanced'`。在 `XGBoost` 中，可以使用 `scale_pos_weight` 參數來平衡正負樣本的權重。

二、算法层面的改进

1. 代价敏感学习（Cost-Sensitive Learning）

加权损失函数：调整正负样本的损失权重。

python

PyTorch示例

`pos_weight = torch.tensor([10.0])` # 正样本权重

`criterion = nn.BCEWithLogitsLoss(pos_weight=pos_weight)`

Focal Loss：降低易分类样本的权重，聚焦难样本。

python

`class FocalLoss(nn.Module):`

`def __init__(self, alpha=0.25, gamma=2):`

`super().__init__()`

`self.alpha = alpha`

`self.gamma = gamma`

`def forward(self, inputs, targets):`

`BCE_loss = nn.BCEWithLogitsLoss(reduction='none')(inputs, targets)`

`pt = torch.exp(-BCE_loss)`

`loss = self.alpha * (1-pt)**self.gamma * BCE_loss`

`return loss.mean()`

2. 模型结构优化

双塔模型（Two-Tower Model）：

用户特征塔和广告特征塔分开处理，最后交互。

减少稀疏特征的直接耦合。

集成学习：

Bagging：对正样本多次采样训练多个模型。

Boosting：迭代调整样本权重（如AdaBoost）。

3. 异常检测思路

将点击预测视为异常检测：使用One-Class SVM或Isolation Forest。

三、评估指标的调整

1. 避免准确率陷阱

在資料不平衡的情況下，僅使用準確度 (Accuracy) 作為評估指標是具有誤導性的。因為即使模型預測所有樣本為多數類，其準確度也會很高。

應該使用以下指標：

- **混淆矩陣 (Confusion Matrix):** 直觀地顯示真陽性 (TP)、真陰性 (TN)、假陽性 (FP)、假陰性 (FN)。
- **精確率 (Precision):** $TP/(TP+FP)$ 。模型預測為點擊的樣本中，有多少是真正的點擊。
- **召回率 (Recall / Sensitivity):** $TP/(TP+FN)$ 。所有真正的點擊中，有多少被模型正確預測。
- **F1-Score:** $2*(Precision*Recall)/(Precision+Recall)$ 。精確率和召回率的調和平均值，綜合考慮了兩者。
- **PR 曲線 (Precision-Recall Curve):** 比 ROC 曲線更適用於高度不平衡的數據集，因為它關注於少數類別的表現。
- **AUC-ROC (Area Under the Receiver Operating Characteristic Curve):** 衡量模型區分正負類別的能力，但在極端不平衡數據下，PR 曲線通常更具信息量。

2. 业务对齐指标

Calibration：确保预测概率与真实点击率一致（如Platt Scaling）。

3. 調整決策閾值 (Adjusting Decision Threshold)

- 在許多分類模型中，預設的分類閾值是 0.5。對於不平衡數據，可以調整這個閾值。例如，如果模型預測的機率高於 0.2 就視為點擊，這可能增加召回率（找到更多點擊），但同時可能降低精確率（更多誤報）。這通常需要根據業務需求和實際情況進行權衡。

RIG (Relative Information Gain)：对比基准模型的提升。

四、工程实践技巧

1. 负样本降采样 (Online Negative Down Sampling, ONDS)

训练时随机丢弃部分负样本（如保留10%），预测时校准输出：

$$p_true = p_pred / [p_pred + (1-p_pred) / w]$$

其中 w 是采样率（如0.1）。

2. 曝光偏差修正

仅使用点击数据：忽略未点击的曝光（适用于部分场景）。

逆倾向加权 (Inverse Propensity Weighting, IPW)：

$$w_i = 1 / [p(\text{show}_i | \text{context})]$$

其中 p 是广告被展示的概率。

3. 增量学习与冷启动

冷启动处理：对新广告/用户使用内容特征（非行为特征）。

在线学习：实时更新模型（如FTRL优化器）。

五、前沿方法

1. 元学习 (Meta-Learning)

MAML：让模型快速适应新广告的稀疏点击模式。

2. 对比学习 (Contrastive Learning)

构建正负样本对，学习更鲁棒的表示。

3. 多任务学习

联合预测点击率和后续转化率（CVR），共享特征表示。

六、方案选择建议

场景	推荐方法
数据量中等 (<100GB)	Focal Loss + 过采样 + AUC优化
超大规模数据	负样本降采样 + 双塔模型
冷启动问题严重	元学习 + 内容特征增强
需要概率校准	Platt Scaling + 代价敏感学习

代码示例 (完整Pipeline)

```
python
import torch
```

```
from sklearn.utils import resample
```

```
# 1. 数据重采样
```

```
def resample_data(X, y, target_ratio=0.3):  
    pos = X[y == 1]  
    neg = X[y == 0]  
    neg_downsampled = resample(neg, n_samples=int(len(pos) * (1-target_ratio)/target_ratio))  
    return torch.cat([pos, neg_downsampled]), torch.cat([torch.ones(len(pos)),  
    torch.zeros(len(neg_downsampled))])
```

```
# 2. Focal Loss模型
```

```
model = nn.Sequential(  
    nn.Linear(input_dim, 256),  
    nn.ReLU(),  
    nn.Linear(256, 1)  
)  
criterion = FocalLoss(alpha=0.25, gamma=2)
```

```
# 3. 训练循环
```

```
for X_batch, y_batch in dataloader:  
    optimizer.zero_grad()  
    outputs = model(X_batch)  
    loss = criterion(outputs, y_batch.float())  
    loss.backward()  
    optimizer.step()
```

通过组合数据重采样、损失函数优化和评估指标调整，可显著提升广告点击预测系统对稀疏正样本的捕捉能力。实际应用中需根据数据规模和业务需求选择合适的方法组合。