

# MIL TRACK

## 一. 算法概述

算法的全名叫做Multiple Instance Learning(多示例学习), 是粒子滤波和boost的结合。算法采用随机产生特征和样本的策略, 通过在线训练弱分类器并更新特征来对实时样本进行在线分类, 所有弱分类器形成的强分类器输出最大者对应的位置即为当前的跟踪结果。

## 二. 算法流程

1. **第一帧进行人脸检测**, 记录矩形框尺度以及左上角点坐标和右下角点坐标
2. **从特征池中随机选择 M 个随机特征**(包括位置、尺度和类型), 具体注意事项见函数 randomMFeatures.m, 采用的是 5 类 haar 特征, A:左右, B:上下, C:左中右, D:对角, E:上中下
3. **在距当前矩形位置左上角点一定距离范围内随机选择负样本和正样本**(只记录左上角点坐标), 见函数 randomSelectSample.m
4. **计算每一个样本对应的 M 个候选弱分类器的输出,以及初始化|更新每个特征对于所有正样本或所有负样本的均值和标准差**, 这一步数据量较大容易出错要仔细写, 见函数 candidateWeakClf.m

函数内的流程解释以及弱分类器形式:

- 1) 计算每个样本对应每个特征的特征值, 保存于变量 AllFeatures
- 2) 以及初始化/更新每个特征对于所有正样本或所有负样本的均值和标准差
- 3) 求弱分类器输出, 弱分类器形式为:

$$h_m(x_n) = \log[p(y_n=1|f_m(x_n)) / p(y_n=0|f_m(x_n))]$$
, 其中  $f_m(x_n)$  表示样本  $x_n$  对应的第  $m$  个特征,  $y_n=1$  为正样本标记,  $y_n=0$  为负样本标记, 按照bayes条件概率公式  $P(B|A)=P(B)P(A|B)/P(A)$  展开, 并让  $p(y_n=1) = p(y_n=0)$ , 得

$$h_m(x_n) = \log[p(f_m(x_n)|y_n=1) / p(f_m(x_n)|y_n=0)]$$

其中  $p(f_m(x_n)|y_n=1)$  服从均值为  $\mu_1$ , 标准差为  $\sigma_1$  的高斯分布,  $p(f_m(x_n)|y_n=0)$  服从均值为  $\mu_2$ , 标准差为  $\sigma_2$  的高斯分布

写成代码的形式为:

$h\_candidateWeakClf = \log((P\_F\_1+0.000000001)/(P\_F\_0+0.000000001))$ ，加 0.000000001 是为了防止 log 无意义的情况发生

## 5. 从 M 个候选弱分类器中选择 K 个最佳弱分类器

每个负样本单独做为一个包，所有正样本做为一个包，计算样本包的概率  $P\_main$ (列向量，行数为：负样本数+1)，里面用到了 S 型函数，所有负样本包标记为 0，正样本包标记为 1，存储于  $Label\_y$ (维数与  $P\_main$  相同)

这个地方用文字叙述比较麻烦，还是见程序吧，思路不复杂，或者我讲讲，要注意的地方如下：

### 1) 计算正样本包的概率时用到 NOISY-OR 模型

$P\_main\_1 = 1 - \prod(1 - p\_each)$ ，其中  $p\_each$  为正样本包中的每个样本的概率，NOISY-OR 模型意思为只要序列中一个元素为 真 则整个序列就为 真，也就是说正样本包中只要有一个样本为真正的正样本则整个正样本包则  $P\_main$  就为真(接近 1 或等于 1)

### 2) 第 m 个特征对应的损失函数(错误率)为：

$$L(m,1)=abs(sum(Label\_y.*log(P\_main+0.000000001) + (1-Label\_y).*log(1-P\_main + 0.000000001),1))$$

## 6. 进入下一帧进行搜索

1) 在距上一帧矩形位置左上角点一定距离范围内随机选择一些候选位置(如 500 个)，采用半固定半随机策略，即在原位置较近区域内固定 100~300 个位置，其余的位置在稍远区域内随机选择，这样可以避免完全随机产生位置而使只有很少一部分位置在上一帧位置的左上角点的邻近区域内。见函数 `randomSelectSample_for_result.m`

2) 将 500 个候选位置对应的样本用 K 个弱分类器(组成一个强分类器)分类，对应强分类器输出最大的位置(矩形左上角点坐标)即为最终结果,见函数 `compute_reult.m`

重复步骤 3~6