# ERT

基础研究组

## 理论

基于待预测形状（landmarks）位于由训练样本形状构成的线性空间的假设，ERT 提出了以随机森林（cascade）作为基本预测数据结构的训练－预测算法来对人脸形状进行预测。ERT 的预测形状由粗略估计和冗余（误差）补偿两部分构成，详细来说，是由训练过程中逐级生成的随机森林由粗到细逐步补偿的。目前，该算法具有预测准确，预测速度快的优点。

ERT 训练得到的模型文件是由多个负责不同预测粒度的随机森林构成的，而随机森林是由初始回归器 $f_0$ 和回归树 $g_k, k \in \{k|1 \le k \le K\}$ 构成的，它们是通过 GBT（Gradient Boosting Tree）算法生成组织起来。由 ERT 训练得到的模型文件如图（1）所示：

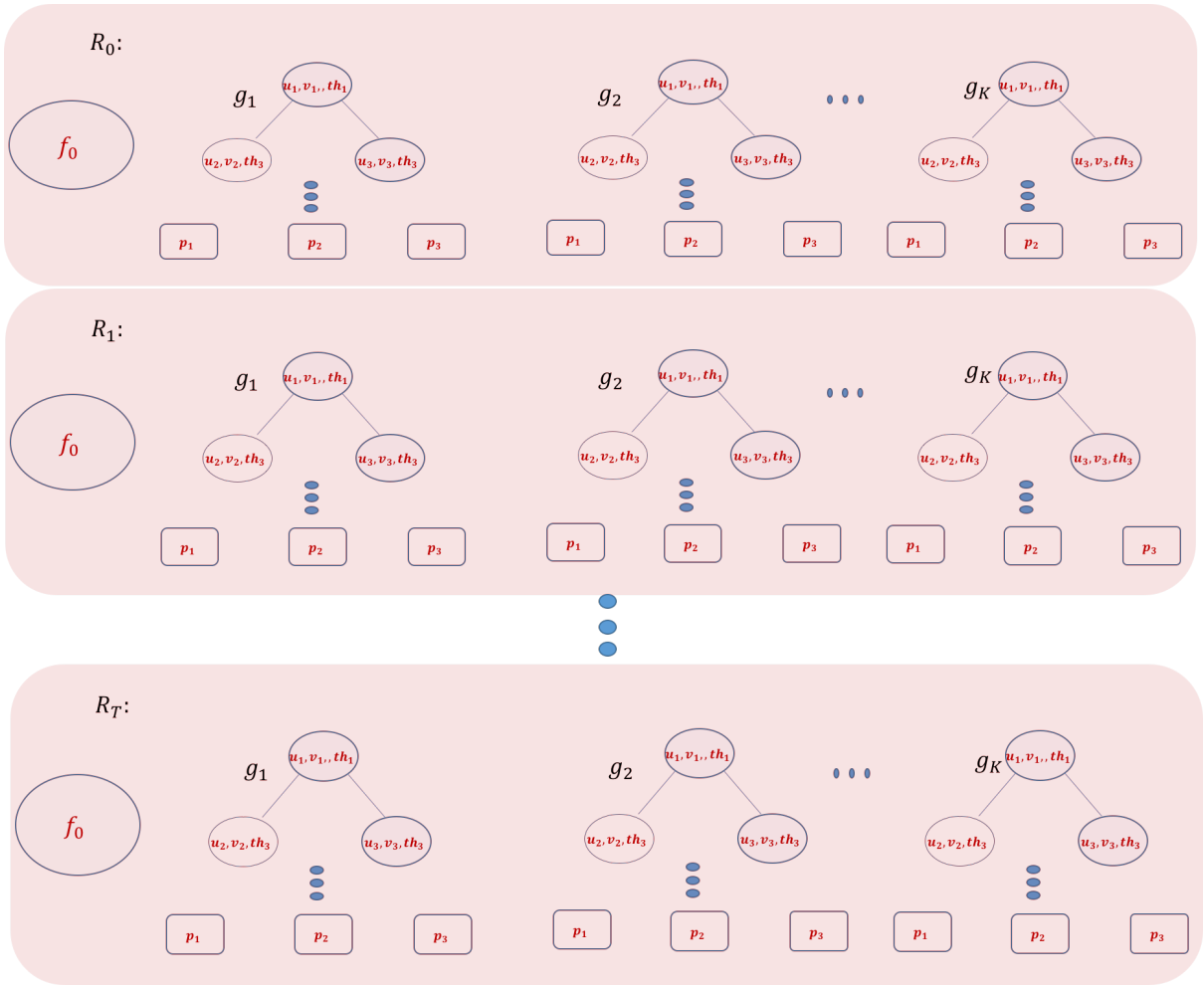


**图 1.** ERT 的模型文件

ERT 就是利用这些初始回归器和回归树来对冗余进行预测的，最终得到的预测形状由公式 1 表示：

$$S_I = S' + \sum_{t=1}^{T} R_t(I) \qquad (1)$$

其中，$S'$为训练样本的平均形状，$R_t(I)$为随机森林对冗余的补偿。在冗余补偿过程中，随机森林$(R_t)$的补偿功能越来越精细。

在训练$R_t$时，首先从样本的真实形状$S_{truth}$中扣除估计形状$S_{estimate}$得到$Residual_0$，然后从$Residual_0$训练得到$f_0$，再从$Residual_0$中扣除由$f_0$预测的冗余估计得到$Residual_1$。重复这个训练一扣冗余的过程，最终得到 K 个回归树。$R_t$对冗余的估计就是由$f_0$和$g_k, k \in \{k|1 \le k \le K\}$的所有冗余估计线性加权得到的，$R_t$的的构成如图 2 所示，冗余估计见图中公式。

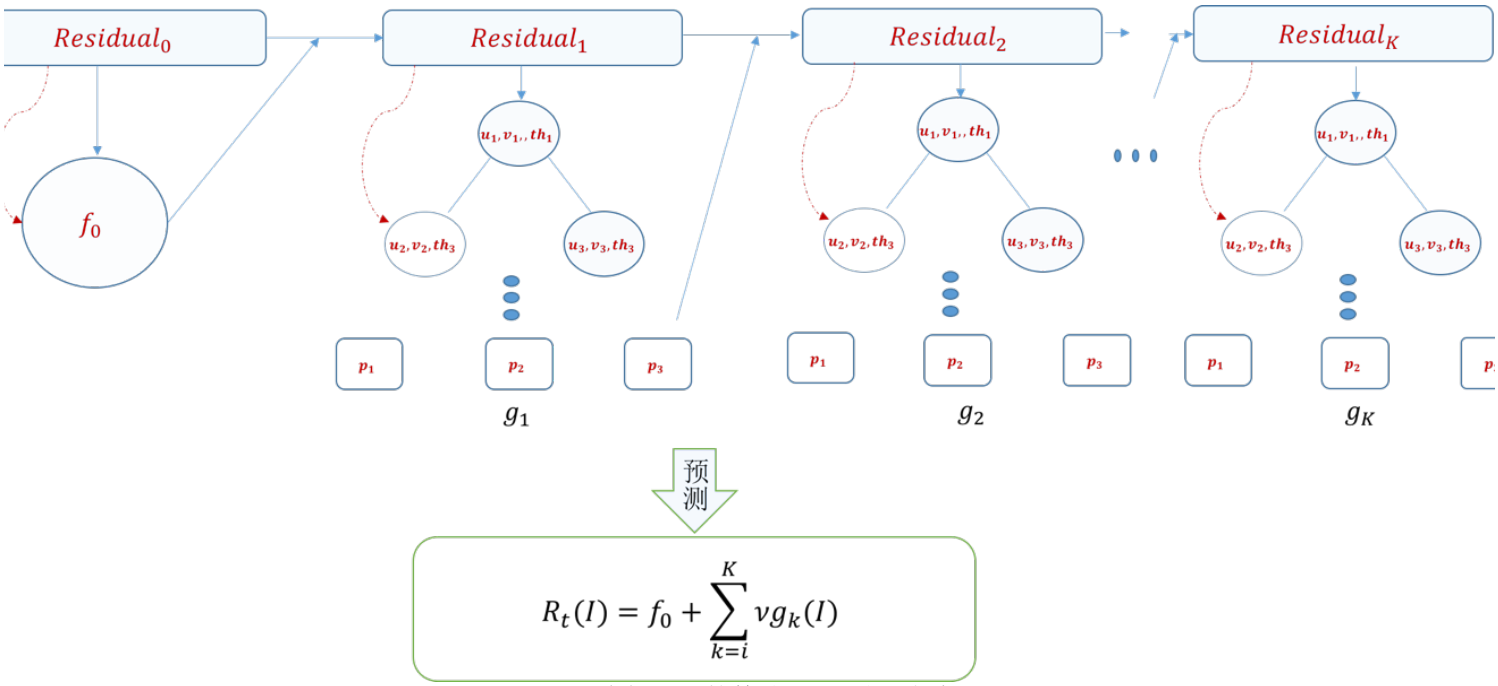


**图 2**. $R_t$的构成以及预测公式

$f_0$和$g_k$对冗余的估计均可由训练样本形状线性表示，而 ERT 的初始粗略估计$S'$也位于由训练样本构成的线性空间，因此最终的预测形状一定位于该线性空间，可以看出 ERT 不需要添加额外的约束条件就可以保证待预测形状位于由训练样本形状构成的线性空间的假设。

在训练具体的一棵回归树时，ERT 将随机点对的强度差作为特征，当强度差小于设定的阈值 *th* 时，训练样本被划入左子树，反之则被划入右子树，到达设定的树深 F 时，计算相应范围内的平均冗余$Residual_k$，即得到冗余估计。从中可看出，回归树就是图像信息与形状信息的映射函数。从另一个角度来看，树的训练过程就是对样本进行分类的过程。在树的建立过程中，ERT 利用公式（2）来保证分类纯度。

$$\mathrm{E}(\mathbb{Q}, \theta) = \sum_{s\epsilon\{r,l\}} \sum_{i\epsilon\mathbb{Q}_{\theta,s}} \|r_i - \mu_s\|^2$$

$$\mu_s = \frac{1}{|\mathbb{Q}_{\theta,s}|} \sum_{i\epsilon\mathbb{Q}_{\theta,s}} r_i$$

(2)

从公式可看出，ERT 是将集合内冗余的波动来作为纯度的度量标准，即波动越小，纯度越高。

## 实现细节

1. 在训练前，先将所有形状标准化，然后进行训练
2. 在训练$R_t$前要随机生成 P（400）对点，每做一次分割就从中随机选取一定数目（≤ S）的点对，从中选择具有最大纯度的分割。
3. ERT 论文的 2.3.1 节形状变换模型（ERT 论文中公式 8）的求解过程如下：

   模型：$\min f = \sum_{j=1}^{P} \left\| X_j' - (s_i R_i X_{i,j} + t_i) \right\|_2$

   求解：因为$s_i$为标量，$R_i = \begin{vmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{vmatrix}$, $t_i = \begin{vmatrix} t_x \\ t_y \end{vmatrix}$

   因此，$s_i R_i X_{i,j} = s_i \begin{vmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{vmatrix} * \begin{vmatrix} x_{i,j} \\ y_{i,j} \end{vmatrix} = \begin{vmatrix} a_x x_{i,j} - a_y y_{i,j} \\ a_x x_{i,j} + a_y y_{i,j} \end{vmatrix}$,其中：$a_x = s_i cos\theta$，$a_y = s_i sin\theta$

   分别对标量：$a_x$，$a_x$，$t_x, t_y$求偏导数，并联立得到如下方程组：

$$\begin{cases} \dfrac{\partial f}{a_x} = 0 \\ \dfrac{\partial f}{a_y} = 0 \\ \dfrac{\partial f}{t_x} = 0 \\ \dfrac{\partial f}{t_y} = 0 \end{cases}$$

   将目标函数代入该方程组并求解可得如下方程组：

$$\begin{cases} \mathrm{A_i} a_x - \mathrm{B_i} a_y + \bar{W} t_x + 0 \cdot t_y = \bar{A} \\ B_i a_x + A_i a_y + 0 \cdot t_x + \bar{W} t_y = \bar{B} \\ Z_i a_x + 0 \cdot a_y + A_i t_x + B_i t_y = C_1 \\ 0 \cdot a_x + Z_i a_y - B_i t_x + A_i t_y = C_2 \end{cases}$$

（3）

其中：$\mathrm{A_i} = \sum_{j=1}^{P} w_j x_{i,j}$, $B_i = \sum_{j=1}^{P} w_j y_{i,j}$, $\bar{W} = \sum_{j=1}^{P} w_j$, $\bar{A} = \sum_{j=1}^{P} w_j \bar{x_j}$, $\bar{B} = \sum_{j=1}^{P} w_j \bar{y_j}$

$C_1 = \sum_{j=1}^{P} w_j (x_{i,j} \bar{x_j} + y_{i,j} \bar{y_j})$, $C_2 = \sum_{j=1}^{P} w_j (x_{i,j} \bar{y_j} - y_{i,j} \bar{x_j})$, $Z_i = \sum_{j=1}^{P} w_j (x_{i,j}^2 + y_{i,j}^2)$

$X_{i,j}: [x_{i,j}, y_{i,j}], j \in [1, P], W = \begin{pmatrix} w^1 & 0 & \cdots & 0 & 0 \\ 0 & w^1 & & 0 & 0 \\ \vdots & & \ddots & \vdots & \\ 0 & 0 & \cdots & w^P & 0 \\ 0 & 0 & & 0 & w^P \end{pmatrix}$

只要P 至少为 1，那么（3）就可以求可解。

## Matlab 代码说明

**testTrain：**

输入：0 代表用 C＋＋，1 代表用 matlab

功能：该文件为训练文件，主要读取数据，设置参数，制作训练样本

输出：0 保存为 predicter_c.mat, 1 保存为 predicter_matlab.mat

注：读取的灰度图像存储在 csv 格式的文件里，读取的形状存储在 asf 格式的文件里

**testPredict：**

输入：0 代表用 C＋＋，1 代表用 matlab

功能：预测一帧人脸图像的形状

**createCascade：**该文件生成指定个数的随机森林

**createTree：**该文件生成一棵树

**splitNode：**该函数可生成树的节点，主要涉及随机选取 w（w≤ S）对分割点，然后从中选去纯度最大的分割

**est：**求形状变换

训练过程中主要操作名为 td 的结构体对象：

```
td =

                imgs: {1x240 cell}
              shapes: {1x240 cell}
               rects: {1x240 cell}
        scaleFactors: {1x240 cell}
                 rnd: 4874080312
      shapeToImages: {1x240 cell}
             samples: {1x4800 cell}
  meanEstimateShape: [2x58 double]
```

imgs：存储训练的真实图像

shapes：存储相应的形状

rects：从形状计算的人脸监测矩形

shapeToImages：为正规化形状到真实形状的形状变换

samples：存储样本数据，主要用于训练

训练后，得到 predicter_xx.mat 文件，该文件主要的数据是 cascades，描述如下：

```
cascades =

    [1x1 struct]    [1x1 struct]

K>> cascades{1}

ans =

           relative: [2x40 double]
    closetLandMarks: [43 32 13 2 22 53 3 17 37 29 33 39 7 50 27 3 24 11 51 44 34 14 44 37 58 12 20 13 8
       meanEstimate: [2x58 double]
                 nu: 0.080000000000000
                  F: 6
               base: [2x58 double]
             trees: {1x10 cell}

K>> cascades{2}

ans =

           relative: [2x40 double]
    closetLandMarks: [34 45 40 38 44 22 49 39 40 51 9 46 14 51 50 54 56 37 1 22 6 36 30 2 37 2 4 5 35 39
       meanEstimate: [2x58 double]
                 nu: 0.080000000000000
                  F: 6
               base: [2x58 double]
             trees: {1x10 cell}
```

cascades{i}就是第 i 个森林，预测时主要用到 base 和 trees 两个数据，base 就是前文$f_0$的预测结果，trees 里面是 K 棵树，叶子节点存储了相应的冗余估计。

## 存在的问题

1. 随着随机森林个数的增多，就会出现很多比较小的相近小数的减法，这个运算很容易造成误差的累积。
2. 论文里的参数未必具有普适性
3. 开发初始阶段，为了保证代码的正确性，有些细节完全按照开源代码来写，如纯度验证和形状变换。经过测试，会有 bug 出现。

## 下阶段工作

解决抖动和模型文件过大的问题。主要从以下几个方面展开：

1. 从特征选取的角度增加一些约束，使得对于微小的移动不过于敏感
2. 由于不同的随机森林修复冗余的粒度是由粗到细的，考虑在保证精度的前提下减少随机森林的个数
3. 查阅文献，调查精度良好的线性整体形状算法，利用该算法调整 ERT 的初始估计，期望 ERT 只做些精细化的冗余补偿
4. 追踪近几年领域内的文献，写综述