

Lecture 5: 特征选择与特征提取

引入

- 特征提取：主成分分析
- 特征选择：自适应提升算法

特征选择和提取简介

概念：特征选择问题

- 概念：
 - 假设有一串向量 $[x_1, x_2 \cdots x_p]$
 - 假设其中的 $X_i = [X_{i1}, X_{i2} \cdots X_{iN}]$
 - 其中的每一个 X_i 属于 C_1 或 C_2
- 问题：
 - N 个维度中有冗余；
 - 如何从 N 个维度当中、选取 M 个维度的特征 ($M < N$)，使得识别率最高；
 - 这其实就是一个特征选择的问题；

概念：特征提取问题

- 前提：
 - 假设有 N 个维度： $[X_{i1} \times_{i2} \cdots X_{iN}]$
- 问题：
 - 要构造函数： $[f_1(x_{i1} - x_{iw}), f_2(x_{i1} - x_{iw}) \cdots f_\mu(x_{i1} - x_{iw})]$ ，即函数的输入是 N 维，输出降到了 M 维；
 - 希望能够降维，并且充分的保留原本的信息；
 - 这种问题叫做特征提取；

特征提取 - 主成分分析

主成分分析基础概念

概念：主成分分析

- 概念：
 - 构造一个 A , b ，使得 $Y = AX + b$;
 - 其中： Y 是 $M \times 1$ ， X 是 $N \times 1$ ；
 - 即：希望构造一个函数，使得原本输入的向量是 N 维，输出的结果降维为 M 维；
- 总结：
 - 主成分分析可以理解为一个一层的、有 M 个神经元的神经网络，即 $Y = W^T x + b$;
- 注意：
 - 主成分是没有标签 (Label) 的；

主成分分析的实现

概念：主成分分析的实现

- 本质：
 - 寻找方差最大的方向，并在该方向投影；
- 概念：方差最大的方向
 - 假设在二维空间中有很多个向量，它们分布在二维空间中。然后主成分分析希望在这个二维空间中找一条直线，使得每个点到这个线段的距离的方差最大；
 - 获得这条线段后、就可以将每个点投影到这个线段，然后得到一个一维的轴，因此用一维的轴、保存了二维的信息；
- 公式：
 - 由算法：

$$\begin{cases} Y = A(X - \bar{X}) \\ \bar{X} = E(X) \\ \bar{X} = \frac{1}{p} \sum_p X_p \end{cases}$$

- 其中：

$$Y_i = \begin{bmatrix} a_1(x_i - \bar{x}) \\ a_2(x_i - \bar{x}) \\ \vdots \\ a_n(x_i - \bar{x}) \end{bmatrix} = \begin{bmatrix} y_{i1} \\ y_{i2} \\ \vdots \\ y_{im} \end{bmatrix} \quad (i = 1 \text{ 到 } p)$$

- 目的：最大化

$$\sum_{i=1}^p (y_{i1} - \bar{y}_{11})^2 = \sum_{i=1}^p a_1 [(x_i - \bar{x})(x_i - \bar{x})^T] a_1^T$$

- $$= a_1 \left[\sum_{i=1}^p (x_i - \bar{x})(x_i - \bar{x})^T \right] a_1^T$$

- 最后得到的就是一个 1×1 的矩阵，也就是一个数；
- 因此也就等于：最大化 $a_1 \sum a_1$
- 其中的限制条件： $a_1 a_1^T = \|a_1\|^2 = 1$ 即单位矢量；
- 求解：使用拉格朗日乘子法

- $$E(a_1) = a_1 \sum a_1^T - \lambda(a_1 a_1^T - 1)$$

- $$\frac{\partial E}{\partial a_1} = \left(\sum a_1^T - \lambda a_1^T \right)^T = 0$$

- 其中偏导就是对 a_1 其中每一个分量求偏导；
- 即： $\sum a_1^T = \lambda a_1^T$ ，其中 a_1^T 是 $\sum$ 的特征向量， λ 是 $\sum$ 的特征值；

- 则：

$$a_1(\sum a_1^T) = a_1(\lambda a_1^T) = \lambda(a_1 a_1^T) = \lambda$$

- 因此 λ 是二最大特征值, a_1 是 $\sum$ 最大特征值对应的特征向量, 且 $a_1 a_1^T = 1$;
- 推广：三维即以上
 - 以上是二维的情况, 如果是三维即以上的情况, 则还需要更多的限制条件:
 - 当还需要最大化 $a_2 \sum a_2^T$ 时, 此时还需限制条件:
 - $a_2 a_2^T = \|a_2\|^2 = 1$
 - $a_2^2 = 1 a_2 a_1^T = a_1 a_2^T = 0$
 - 此时继续使用拉格朗日乘子法:
 - $E(a_2) = a_2 \sum a_2^T - \lambda(a_2 a_2^T - 1) - \beta a_1 a_2^T$

PCA 算法

概念： PCA 算法过程

- (1) 求协方差矩阵:

$$\sum = \sum_{i=1}^p (X_i - \bar{X})(X_i - \bar{X})^T$$

- (2) 求协方差矩阵的特征值, 并按照从大到小排序, 得到:
 - $[\lambda_1, \lambda_2 \dots, \lambda_N, \lambda_{N+1} \dots]$
 - 对应的特征向量为:
 - $[a_1, a_2 \dots a_n, a_{n+1}, \dots]$
- (3) 归一化所有的 a_i , 使得所有的 $a_i a_i^T = 1$
- (4) 将所有的 a_i 拼成一个矩阵 A ;
- (5) 解得 $Y_i = A(X_i - \bar{X})$

主成分分析的应用

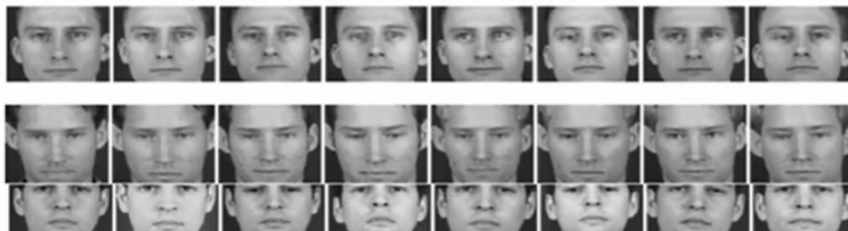
分析： 主成分分析的应用

- 例子:

数据库：XM2VTS，由295人构成，每人8张照片，分四次拍摄，每次两张，每次间隔时间一个月。



人脸检测算法截取人脸后，大小归一化到128*128像素



- 这里的 X 也就是图片，是 128×128 ；

对每一个人，用前两次拍摄4张图片训练，用后两次拍摄4张图片测试。

训练数据：295*4=1180张， 测试数据：1180张

做PCA后，取前100维向量进行测试

	最近邻法	SVM
识别率	76.02%	80.25%

- 因为是取 100 个维度，因此是 $128 \times 128 \times 100$ ，输出的 Y 是 100×1 ，即输出是 100 维的向量；
- 拿这 100 维的向量的数据拿去做特征提取，使用两种方法；



- 特征脸就是图像中最具有特征的那些部分；

特征选择 - 自适应提升

特征选择的问题

概念：特征选择的问题

- 概念：
 - 从 M 个特征中选择 N 个使得识别率最高；

- 选择的可能个数: $C_N^M = \frac{N!}{M!(N-M)!}$
- 因此如果暴力的破解, 选择, 则复杂度十分大;
- 方法:
 - 以下是启发式算法;
 - 1. 递增法: x_1, x_2, x_3
 - 2. 递减法: x_3, x_2, x_1
 - 但是这些方法并不好用, 因为神经网络已经做了这些事情;
- 解决:
 - 使用自适应提升算法

AdaBoost

概念: AdaBoost

- 定义:
 - 假设数据集为 $T = (x_1, y_1, \dots, (x_n, y_n))$;
 - 当前是一个二分类问题, 即 $y_i = (-1, +1)$
- 思想:
 - 先选择一个特征, 用这个特征将数据集分开。这种分开的分类效果只比瞎猜好一点点, 因此是一个弱分类器;
 - 这样做的结果是一定有一些分类错的结果、分类对的结果;
 - 接下来, 更多的采集哪些分类错的样本, 更少的采集哪些分类对的样本, 这样得到下一个数据集;
 - 而这样得到的数据集的特点是: 有一些分类对了的样本可能没有被采样到, 但是那些上一次分类错的样本可能会被再次采集;
 - 此时对新采集的数据集里面、再进行一次评估, 得到一个识别器, 也是一个弱识别器;
 - 然后进行第三次, 此时对于那些识别器分类错的样本、赋予更大的权重, 对于那些分类对了的样本、赋予更小的权值。因此再进行采集时, 那些分类错的样本更多的被采集;
 - 重复上面的过程, 得到加权平均值;
- 流程:
 - 输入: $T = (x_i, y_i)_{i=1 \text{ 到 } N}$
 - 输出: 分类器 $G(x) = +1 \text{ 或 } -1$
 - (1) 初始化采样权值:
 - $D_1 = (W_{11}, W_{12}, W_{13}, \dots, W_{1N})$
 - $W_{1i} = \frac{1}{N} \quad (i = 1 - N)$
 - (2) 对 $m = 1, 2, 3 \dots M$ (M 是分类器的个数)
 - 用 D_m 采样 N 个训练样本, 在训练样本上获得分类器, 得到 $G_m(x) = +1 \text{ 或 } -1$
 - 这是一种有放回的采样
 - (3) 计算加权错误率
 - 加权错误率:

$$e_m = P(G_m(x_i) \neq y_i) = \sum_{i=1}^N w_{m1} I(G_m(x_i) \neq y_i)$$

- 识别器的权重:

$$a_m = \frac{1}{2} \log \frac{1 - e_m}{e_m}$$

- (4) 更新权值分布

$$D_{m+1} = (W_{m+1,1}, W_{m+1,2} \dots W_{m+1,N})$$

- (5) 回到 (2) , 循环 M 次
- (6) 最终识别器 $G(x)$:

$$f(x) = \sum_{n=1}^M a_n G_n(x)$$

$$G(x) = \text{sign}(f(x)) = \text{sign}\left[\sum_{n=1}^M a_n G_n(x)\right]$$

获取笔记持续更新可加微信: Wandu2277