# 论文阅读笔记（1）

——《Fully Automatic 3D Facial Expression Recognition using Local Depth Features》

## 概述

论文信息

作者：Mian, A., W. Liu, L. Li, and M. Xue

会议：WACV 2014: IEEE Winter Conference on Applications of Computer Vision, Mar 24, 2014

论文实验数据来源：BU-3DFE

论文涉及到的知识点：

|  |  |
| --- | --- |
| 1 | Concentric Geodesic Rings |
| 2 | Haar级联分类器 |
| 3 | AdaBoost |
| 4 | 迭代最近点算法 Iterative Closest Points |
| 5 | 2DPCA |
| 6 | mRMR |
| 7 | 奇异值分解SVD |
| 8 | 互信息 |
| 9 | SVM分类器 |

重点参考文献

|  |  |  |
| --- | --- | --- |
| 1 | 参考文献11 - Feature selection based on mutual information criteria of max-dependency, maxrelevance, and min-redundancy | mRMR |
| 2 | 参考文献19 - Twodimensional pca: a new approach to appearance-based face representation and recognition | 2DPCA |

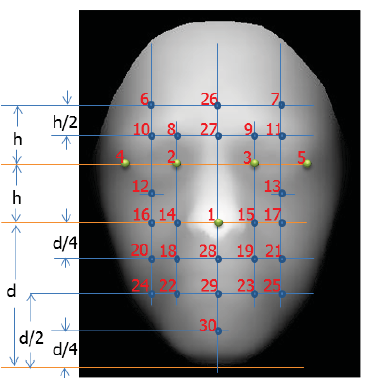
论文贡献

解决了在一幅3D图像中的独立人脸表情识别

表情识别的方法是全自动的

## 预处理

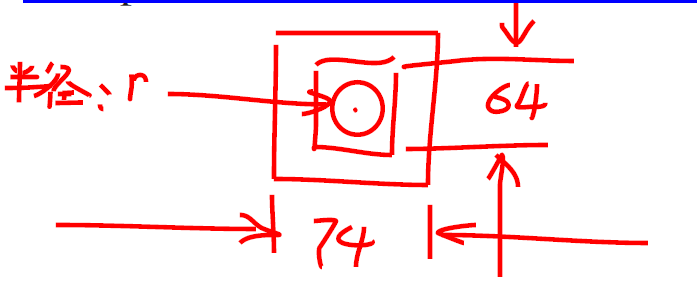
首先找5个基准点（鼻尖 + 4个眼角），再根据一定规则找到25个启发点，一共30个点



## 特征抽取

### 基于面片的深度特征抽取（从30 \* 4096到30 \* 50）

对于每一个点（5个基准点 + 25个启发点），利用一个公开的matlab工具包（文中给的地址无效：<http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do>，最接近的有效地址：<http://www.mathworks.com/matlabcentral/fileexchange/8998-surface-fitting-using-gridfit>）获取一个patch，patch的获取思想如下图所示。



上图中，内方框的边长为64，64 \* 64就是要构建矩阵的数据，它包含有4096个点，每个点的深度信息作为矩阵元素，需要注意的是，尽管64 \* 64看起来是一个矩阵的形态，但实际上进行数据处理时，**将这4096个元素当作一个列向量来构建**的。文中所说的深度特征（Depth Feature）究竟是什么？是模型库中该点的深度数据？这个特征是不是一个独立的数字？还是一个复杂的结构体？

一个训练样本可以理解为一张人脸，每个训练样本都可以用一个 30 \* 4096 的矩阵来表示，30表示有30个点（5个基准点 + 25个启发点），4096 = 64 \* 64（patch的内层区域为64px \* 64px）

现在的问题是：4096维数太高，需要降维，并且相邻的patch可能还存在重叠。降维的基本思想：**利用2DPCA将矩阵投射到一个线性子空间中**。降维的目的是：剔除冗余信息。

降维方法：对每一个样本，构造一个scatter matrix

|  |
| --- |
| 什么是scatter matrix？（散点图矩阵/散点矩阵/散布矩阵）  Given *n* samples of *m*-dimensional data, represented as the *m*-by-*n* matrix （每个样本是一个*m*维的列向量）  计算机生成了可选文字: X=[x1,xZ，…，xn]  the sample mean is  计算机生成了可选文字:  The scatter matrix is the *m*-by-*m* positive semi-definite matrix  计算机生成了可选文字: х  where T denotes matrix transpose, and multiplication is with regards to the outer product. The scatter matrix may be expressed more succinctly as  计算机生成了可选文字: S=XCnXT  where *Cn* is the n-by-n centering matrix.  Cn里面的元素是如何构成的？ |

论文中的散点图矩阵（Scatter Matrix）构造如下，与上面几乎是一样的，只是在前面多除以了一个*N*（*N*表示样本数量），估计是做归一处理，另外标记的字母表示不同

计算机生成了可选文字:
Σ(ΑΙ — — Α). 

其中A表示某一个样本对应的矩阵，其维度为30 \* 4096，C就是散点图矩阵

在计算出了散点图矩阵之后，就要进行**2DPCA**处理，

PCA有什么作用？2DPCA有什么用？两者的区别是什么？

计算机生成了可选文字:
双x)=xTcx

|  |
| --- |
| “PCA一般都是先把样本矩阵变成一维的向量，然后很多样本的向量构成一个矩阵，接下来求协方差矩阵的特征向量等等。  而2DPCA不需要先把样本矩阵变成一维的向量，**直接由数据矩阵构造散布矩阵，一般来说，2DPCA提取的特征比PCA更好，更快，并且计算时间降低**。2DPCA的唯一不足是计算过程中需要存储的系数比较多。  具体的你可以看看Jian Yang发表在PAMI上面的那篇文章。” |

在上面这个公式中，*X*是列向量

最优的标准：使矩阵*C*的特征向量能够符合最大的特征值。How to do?选择一组正交投影向量：*X1, X2, …, Xd*，如何得到投影向量？可以对散点图矩阵*C*进行SVD奇异值分解

计算机生成了可选文字:
c usvT 

其中*U*的大小是4096 \* 4096，S是特征值对角矩阵，矩阵*U*的前 *d*列就是最优投影向量，d怎样求？前*d*个特征所占的比例是这样的：

计算机生成了可选文字:
Σι=1 λι 

其中是第i个特征值。实验表明，当*d* = 50的时候，的值大约可以到0.99，因此，***d* = 50**个特征向量可以选作为最优投影矩阵。

利用刚刚得到的最优投影矩阵可以对样本进行压缩，压缩方法如下：

C:\Users\DISCOV~1\AppData\Local\Temp\msohtmlclip1\02\clip_image012.png

F就是压缩后的矩阵，它的维度是30 \* 50，这样就实现了降维。F可以理解为压缩后的样本。F能不能理解为是抽取后的特征？

### 特征选择

mRMR（minimal-redundancy-maximal-relevance，最小冗余最大相关性），详情见参考文献11

mRMR实施步骤：

找一个特征集S，满足：S中的特征都是互斥的，并且这些特征对六类基本表情（c）具有**最大的判断力**

其中I(xi, c)表示S中的特征xi和表情类别c之间的互信息

计算机生成了可选文字:
I (Xi ; c), 
Xi J 

I(xi, xj)表示S中两个特征之间的互信息。

从感觉上看，就像是希望某个特征xi和表情类别之间有非常紧密的联系，而这个特征和其他特征之间又没有什么关系。

利用SVM分类器对S中的特征进行测试，确定S的大小（即：究竟需要多少个特征最优？）

利用十折交叉验证来选取特征

|  |
| --- |
| 十折交叉验证，用来测试精度。是常用的精度测试方法。将数据集分成十分，轮流将其中9份做训练1份做测试，10次的结果的均值作为对算法精度的估计，一般还需要进行多次10倍交叉验证求均值，例如10次10倍交叉验证，更精确一点。 |

每一次，找一个特征集Si，Si含有800个特征，这800个特征按照分类能力进行降序排列

将所有中（*i* = 1, 2, 3, ... , 10）公共的特征拿出来，放到一起，就形成了里面存放的是候选特征。

为了确定最优特征数量m，从Si中选出50k个特征，送往分类器（其中k是迭代次数，如果需要6次迭代，就选出300个特征），轮番测试。将效果最好的一次测试对应的作为最优特征集，下面是测试记录

计算机生成了可选文字:
Number of selected features 

## 分类

采用SVM进行分类，理由：在综合学习中，SVM算法的性能最优异

## 后续计划

参考文献08 - Shape analysis of local facial patches for 3d facial

参考文献09 - Robust realtime feature detection in raw 3d face images

参考文献11 - Feature selection based on mutual information criteria of max-dependency, maxrelevance, and min-redundancy

## 附1：参考文献11《Feature Selection Based on …》阅读简要笔记

本文的主要工作：

1. 从理论上证明**mRMR算法**的有效性

2. 将**mRMR**与其他特征选择算法相结合

3. 实验对比

特征选择问题的实质：给定N个样本、M个特征，从M维的观察空间、m个特征的子空间当中，找到最优的特征c。

最优分类通常意味着最少的分类错误。在无监督学习的情形下，人们并没有指定分类器，最少错误通常意味着：数据在子空间的分布中，目标类别*c*拥有最大的统计独立性。这种方案就是最大依赖（maximal dependency）。

一种流行的实现最大依赖性的方法是：最大关联特征选择，即选择与类别*c*相关程度最大的特征，关联性通常由关联度或互信息来表示，其中互信息使用很广泛。

给定两个随机变量x, y，它们的互信息是这样的：



在最大相关性理论中，选择出来的特征与类别*c*具有最大的互信息，这反映了一种最大的依赖。

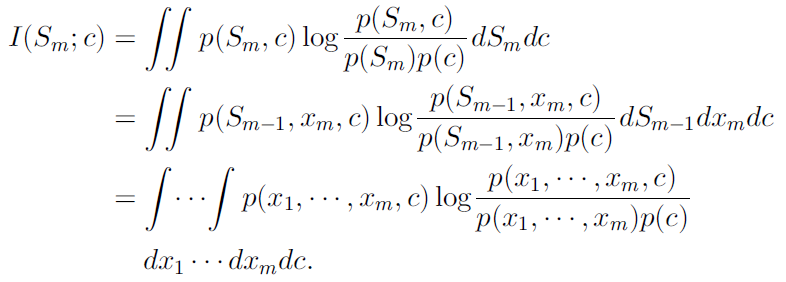
需要注意的是，在特征选择中，将单独的好的特征结合起来并不意味着分类效果也一定好（这里是讲分类效果还是分类的性能？原文表述是classification performance），有些研究人员开始尝试减少特征的冗余，并且根据最小冗余原则来进行特征选择

在互信息理论体系中，特征选择的目的在于：找到一个含有*m*个特征的特征集合*S*，*S*当中所有的特征对目标类别*c*具有最大的依赖，这被称之为**最大依赖**（Max-Dependency）



公式中的标记D应该是对应Dependency，有依赖的意思在里面。*D*用互信息的值来表示，*m*表示特征的数目。

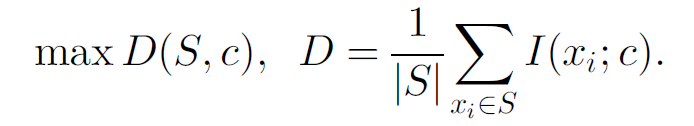
从公式可以看出，当*m*等于1时，解决问题的方法就是寻找一个特征，使其与类别*c*的互信息最大。当*m*大于1时，一个简单的增量搜索方案是：每一次增加一个特征，问题演变成：给定一个集合，它包含*m-1*个特征，那么第*m*个特征可以这样来进行选择：寻找一个特征，它能使具有最大的增长。与特征*c*之间的互信息可以这样求：



抛开最大依赖的理论值不谈，我们很难对多元概率密度进行准确的估计，比如：，或。这是因为在高维空间中存在两个难点：1.样本数不充分；2.多元概率密度估计通常需要计算高维协方差矩阵的转置，而这是不适合的。最大依赖理论还有一个不足，就是计算速度慢。这些问题在**持续特征变量**（continuous feature variables）中表现得最为突出。

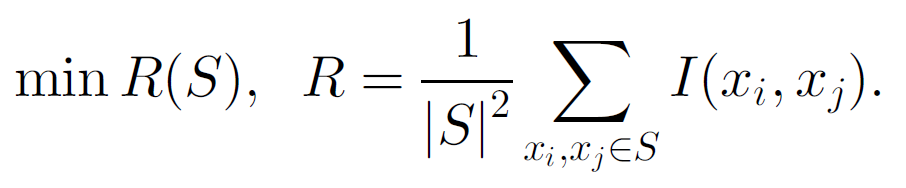
对离散特征而言，在实现最大依赖的过程中，上述问题并不能完全避免。举例来说，假设每个特征有三个类别，有N个样本，K个特征总共有最多种组合情况（原文：joint states），当组合情况的数目上升非常迅速，并且这个数目和样本数量N接近时，这些特征的联合概率和互信息就不能被准确估计。因此，尽管当N很大时，基于最大依赖的特征选择可能会很有用，但是对于那些旨在高分类精度并且with相当紧凑的特征（原文：with a reasonably compact set of features）

由于最大依赖理论难以实现，一种合理的办法是基于**最大相关**（maximal relevance）进行特征选择，**最大相关算法在搜索特征时接近于最大依赖算法**，并且维持了独立特征和类别c之间互信息的平均值，公式如下。



为什么最大相关算法在搜索特征时接近于最大依赖算法？

根据最大相关算法选择出来的特征可能存在大量冗余，也就是说，这些特征之间的依赖会非常大。当两个特征之间的依赖非常大的时候，它们各自的分类能力不会有太大改变if one of them were removed，这句话的意思就是：冗余太大了，移走一两个特征并不是什么大不了的事情。因此，可以加上最小冗余的限制条件，公式如下。



公式中的R应该对应Redundancy，表示冗余的意思。

将上面两个原则：**最大相关**和**最小冗余**结合起来，就是mRMR