## Fisher算法

人脸图像识别任务，其实是多分类任务，在多分类中我们最容易想到也是最简单的方法就是判别法；

距离判别法首先根据已知分类的数据，分别计算出各类的重心。再根据新个体到每类的距离，根据最短的距离确定分类情况。在距离判别法中我们最常用的是欧式距离，当然还有马氏距离等。

为了方便说明问题，在这里我们假设有两个类别和，每个样本测量p个指标特征，即每个样本为，

下面采用欧式距离计算有：

其中； 表示第i类的重心（也就是均值），表示第i类第k个指标的重心。

若采用马氏距离计算，则有

其中； 表示第i类的重心，表示第i类的协方差矩阵的逆矩阵。

基于上述方法我们就能够得出了对于新的样本是属于还是；

如前文所述，人脸识别任务也可以看成是图像的分类任务，故我们可以合理地想到，人脸识别也可以使用距离判别法；

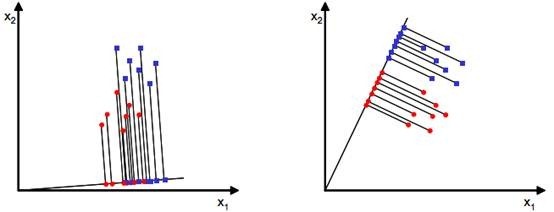
为了验证如上说法，我们进行了使用欧式距离来识别人脸的实验；在本算法中我们统一使用“The Japanese Female Facial Expression (JAFFE) Database”数据集来进行测试和验证，该数据集有十个人的脸（全为女性），每个人有二十多张不同表情的人脸图片；

我们使用欧式距离来实现对该数据集的识别，得到精确度为30%，而没有任何算法和经验的情况下按概率论来说精确度应为1/10，从该实验看到通过距离判别法确实能够一定程度上地进行人脸识别，但是这个结果并不如人意。尽管如此，我们还是看到了通过距离判别从而实现人脸识别的希望。

在《信号与系统》和《自动控制原理》中，一些在时域里难以解决的问题我们通常可以通过将其换到频域中来研究，从而大大减轻了研究的难度。带着同样的思想，我们是否也能通过转换人脸图片的维度从而使距离判别的精确度得到提升呢？显然是可以的，在计算机视觉中，我们不难发现一个规律，无论是传统算法，还是如日中天的神经网络，他们的计算过程都是对图片进行降维，只是在不同方面上的降维罢了。在神经网络中，其实我们可以发现降维的过程会大大地影响着我们模型的最终结果，在降维的过程中，降维越慢越细腻精确度就会越高，每个卷积核对应着一种映射关系和本文即将提到的fisher算法中的k其实是一个道理，在这里不予深究。

基于降维的思想，我们来对图像进行降维，在图像中我们往往有很多像素点，每个像素点代表着图像的一个维度。显然在投影中我们选择其中任何一个像素点也就是特征作为投影的向量在绝大部分的情况下都是十分不明智的选择，因为这样就意味着我们会损失其他的所有特征了。为此我们必须要找到一个合适的投影向量才能进行下一步的工作；我们以简单的两个特征为例展开分析；

下面假设有二维平面上的两个点集x（x是包含横纵坐标的二维向量），它们的分布如下图（分别以蓝点和红点表示两个类别的数据）：



我们现在要找到一个合适的向量w（和数据相同维数），将数据投影到w上（会得到一个标量，直观的理解就是投影点到坐标原点的距离），根据投影点来表示和区分原有数据。以数学公式给出投影点到到原点的距离：。上图中给出了两种w方案，w以从原点出发的直线来表示，直线上的点是原数据的投影点。通过直观判断，不难发现右侧的w更加合理，其上的投影点能够合理的区分原有的两个数据集。下面我们通过计算来得到这个w。

首先计算每类数据的均值（中心点）：

这里的是数据的分类个数， 代表某个分类下的数据点数，比如代表红点的中心，代表蓝点的中心。数据点投影到w上的中心为：

为了得到最佳的w，可以从两方面考虑：1、不同的分类得到的投影点要尽量分开；2、同一个分类投影后得到的点要尽量聚合。从这两方面考虑，可以定义如下公式：

J(w)代表不同分类投影中心的距离，它的值越大越好。

上式称之为散列值（scatter matrixs），代表同一个分类投影后的散列值，也就是投影点的聚合度，它的值越小代表投影点越聚合。

结合两个公式，第一个公式做分子另一个做分母：

上式是w的函数，值越大w降维性能越好，所以下面的问题就是求解使上式取最大值的w。把散列函数展开：

可以发现除w和外，剩余部分可以定义为：

定义：

得到：

展开J(w)的分子并定义SB；

得到最终J(w):

上式求极大值可以利用拉格朗日乘数法，不过需要限定一下分母的值，否则分子分母都变，不妨令，利用拉格朗日乘数法得到：

两边同时乘以可以得到：

不难发现w就是矩阵的特征向量。另外由于

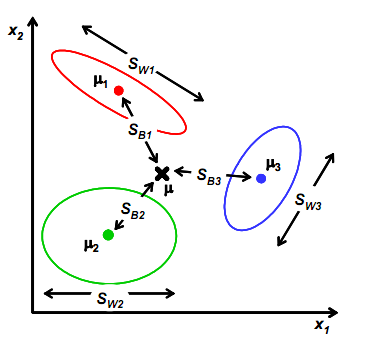
将其带入下式中有：

代回有：

由于向量长度并不影响结果，于是消去和 ，得到

以上就是fisher算法的整个过程，但是上面的计算只能够得出二分类问题的答案，即只能够判断两个人脸。而往往实际中我们需要从多个人的人脸图像中判别该人脸属于哪一个人，于是我们需要将上述算法转换成多分类问题算法，从而实现人脸识别。

假设有C个人的人脸图像，每个人可以有多张图像，所以按人来分，可以将图像分为C类，在分类之前需要处理一下图像，将每张图像按照逐行逐列的形式获取像素组成一个向量，和第一节类似设该向量为x，设向量维数为n，设x为列向量（n行1列）。和之前的二分类不一样，这里的n有可能成千上万，比如100x100的图像得到的向量为10000维，所以第一节里将x投影到一个向量的方法可能不适用了，比如下图：



平面内找不到一个合适的向量，能够将所有的数据投影到这个向量而且不同类间合理的分开。所以我们需要增加投影向量w的个数，设w为：

w1、w2等是n维的列向量，所以w是个n行k列的矩阵。x在w上的投影可以表示为：

我们将从投影后的类间散列度和类内散列度来考虑最优的w, μi代表类别i的中心，而Sw定义如下：

其中：

代表类别i的类内散列度，它是一个nxn的矩阵。

所有x的中心μ定义为：

类间散列度定义：

代表的是每个类别到μ距离的加和，注意Ni代表类别i内x的个数，也就是某个人的人脸图像个数。

上面的讨论都是投影之间的各种数据，而J(w)的计算实际是依靠投影之后数据分布的，所以有：

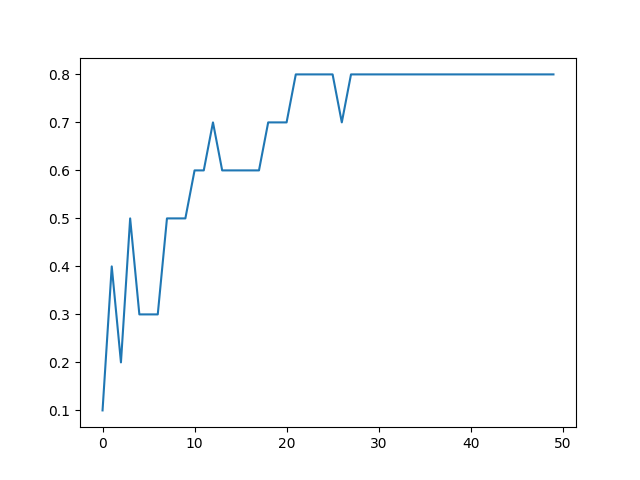
分别代表投影后的类别i的中心，所有数据的中心，类内散列矩阵，类间散列矩阵。

给出如下定义：

得到：

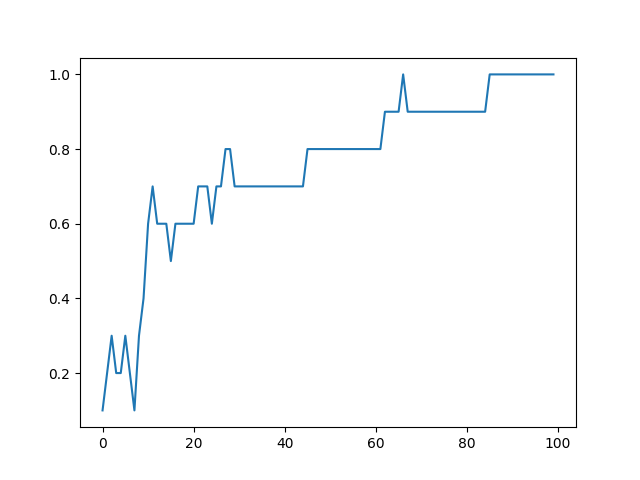
最后得到：

在实验中我们通过使用Fisher算法并采取不同的k值（即不同数量的w），图像大小为20\*20，我们得到如下实验结果



由此可知在k>30时，预测准确率基本稳定在80%；

当图像大小编程50\*50时，结果如下：



在k>90时，精确度可以达到100%；