## 一.基本介绍

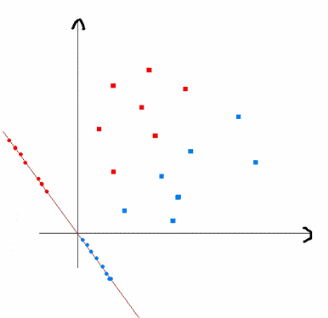
LDA的全称是Linear Discriminant Analysis（线性判别分析），**是一种supervised learning。**有些资料上也称为是Fisher’s Linear Discriminant，因为它被Ronald Fisher发明自1936年，Discriminant指不需要去通过概率的方法来训练、预测数据，比如说各种贝叶斯方法，就需要获取数据的先验、后验概率等等。LDA是在**目前机器学习、数据挖掘领域经典且热门**的一个算法。

 LDA的原理是，将带上标签的数据（点），通过投影的方法，投影到维度更低的空间中，使得投影后的点，会形成按类别区分，一簇一簇的情况，相同类别的点，将会在投影后的空间中更接近。要说明白LDA，首先得弄明白线性分类器([Linear Classifier](http://en.wikipedia.org/wiki/Linear_classifier))：因为LDA是一种线性分类器。对于K-分类的一个分类问题，会有K个线性函数：

[image](http://images.cnblogs.com/cnblogs_com/LeftNotEasy/201101/201101081455463969.png)

     当满足条件：对于所有的j，都有Yk > Yj,的时候，我们就说x属于类别k。对于每一个分类，都有一个公式去算一个分值，在所有的公式得到的分值中，找一个最大的，就是所属的分类了。

    上式实际上就是一种投影，是将一个高维的点投影到一条高维的直线上，LDA最求的目标是，给出一个标注了类别的数据集，投影到了一条直线之后，能够使得点尽量的按类别区分开，当k=2即二分类问题的时候，如下图所示：

[](file:///C:\Documents%20and%20Settings\Administrator\Local%20Settings\Temp\WindowsLiveWriter-429641856\supfiles4CEE5\image%5b15%5d.png)

     红色的方形的点为0类的原始点、蓝色的方形点为1类的原始点，经过原点的那条线就是投影的直线，从图上可以清楚的看到，红色的点和蓝色的点被原点明显的分开了，这个数据只是随便画的，如果在高维的情况下，看起来会更好一点。下面我来推导一下二分类LDA问题的公式：

     假设用来区分二分类的直线（投影函数)为：

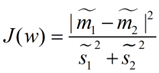
[image](http://images.cnblogs.com/cnblogs_com/LeftNotEasy/201101/201101081455476902.png)

    LDA分类的一个目标是使得不同类别之间的距离越远越好，同一类别之中的距离越近越好，所以我们需要定义几个关键的值。

    1.类别i的原始中心点为：（Di表示属于类别i的点)[image](http://images.cnblogs.com/cnblogs_com/LeftNotEasy/201101/201101081455476869.png)

    2.类别i投影后的中心点为：[image](http://images.cnblogs.com/cnblogs_com/LeftNotEasy/201101/201101081455473248.png)

    3.衡量类别i投影后，类别点之间（即类内）的分散程度（方差）为：[image](http://images.cnblogs.com/cnblogs_com/LeftNotEasy/201101/20110108145548391.png)

    4.最终我们可以得到一个下面的公式，表示LDA投影到w后的损失函数：[](http://images.cnblogs.com/cnblogs_com/LeftNotEasy/201101/201101081455487850.png)（对于两类来说）

   我们分类的目标是，使得类别内的点距离越近越好（集中），类别间的点越远越好。分母表示每一个类别之间的方差之和，方差越大表示一个类之间的点越分散，分子为两个类别各自的中心点的距离的平方，我们最大化J(w)就可以求出最优的w了。只用分母部分内容，会出现如下图1所示的情况：

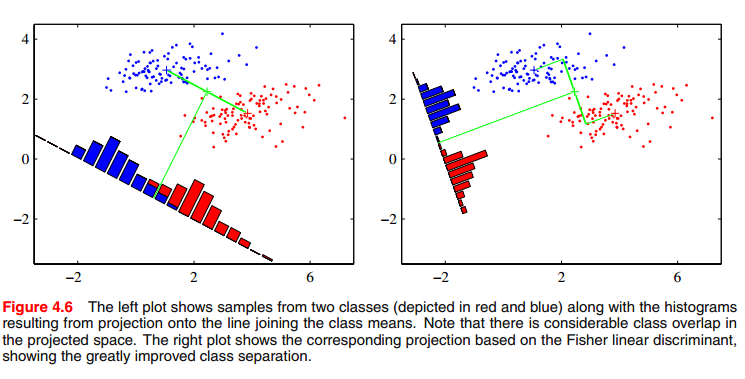


图1 来自 Pattern recognition and machine learning

想要求出最优的w，可以使用拉格朗日乘子法，但是现在我们得到的J(w)里面，w是不能被单独提出来的，我们就得想办法将w单独提出来。

   我们定义一个投影前的各类别分散程度的矩阵，这个矩阵看起来有一点麻烦，其实意思是，如果某一个分类的输入点集Di里面的点距离这个分类的中心店mi越近，则Si里面元素的值就越小，如果分类的点都紧紧地围绕着mi，则Si里面的元素值越更接近0.

[image](http://images.cnblogs.com/cnblogs_com/LeftNotEasy/201101/201101081455491720.png)

   带入Si，将J(w)分母化为：

[image](http://images.cnblogs.com/cnblogs_com/LeftNotEasy/201101/201101081455497751.png)

image

   同样的将J(w)分子化为：

[image](http://images.cnblogs.com/cnblogs_com/LeftNotEasy/201101/201101081455509636.png)

   这样损失函数可以化成下面的形式：

[image](http://images.cnblogs.com/cnblogs_com/LeftNotEasy/201101/201101081455509047.png)

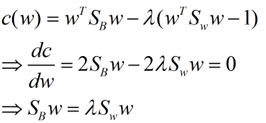
其中SB为类之间的协方差矩阵。SW为类内的协方差矩阵。

对于二元分类：





   这样就可以用最喜欢的拉格朗日乘子法了，但是还有一个问题，如果分子、分母是都可以取任意值的，那就会使得有无穷解，我们将分母限制为长度为1（这是用拉格朗日乘子法一个很重要的技巧，在下面将说的PCA里面也会用到，如果忘记了，请复习一下高数），并作为拉格朗日乘子法的限制条件，带入得到：

[](http://images.cnblogs.com/cnblogs_com/LeftNotEasy/201101/201101081455513997.png)

   其中用到了矩阵微积分，求导时可以简单地把[clip_image061](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104212325024233.png)当做[clip_image063](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104212325029216.png)看待。

 如果[clip_image043[1]](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104212325037198.png)可逆，那么将求导后的结果两边都乘以[clip_image065](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104212325048768.png)，得

[clip_image066](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104212325074817.png)

这个可喜的结果就是w就是矩阵[clip_image068](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104212325073389.png)的特征向量了。

 这个公式称为Fisher linear discrimination。

     等等，让我们再观察一下，发现前面[clip_image070](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104212325089734.png)的公式

[clip_image072](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104212325089145.png)

     那么

[clip_image074](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104212325083256.png)

     代入最后的特征值公式得

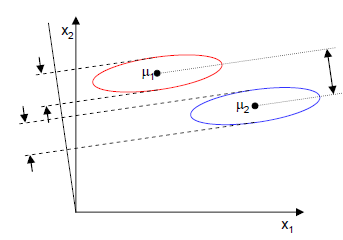
[clip_image076](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/20110421232509715.png)

     由于对w扩大缩小任何倍不影响结果，因此可以约去两边的未知常数[clip_image078](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/20110421232509126.png)和[clip_image080](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104212325103648.png)，得到

[clip_image082](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104212325101107.png)

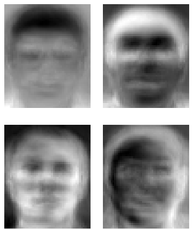
     至此，我们只需要求出原始样本的均值和方差就可以求出最佳的方向w，这就是Fisher于1936年提出的线性判别分析。

     看上面二维样本的投影结果图：

[](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104212325103582.png)

   这里想多谈谈特征值，特征值在纯数学、量子力学、固体力学、计算机等等领域都有广泛的应用，特征值表示的是矩阵的性质，当我们取到矩阵的前N个最大的特征值的时候，我们可以说提取到的矩阵主要的成分（这个和之后的PCA相关，但是不是完全一样的概念）。在机器学习领域，不少的地方都要用到特征值的计算，比如说图像识别、pagerank、LDA、还有之后将会提到的PCA等等。

   下图是图像识别中广泛用到的特征脸（eigen face），提取出特征脸有两个目的，首先是为了压缩数据，对于一张图片，只需要保存其最重要的部分就是了，然后是为了使得程序更容易处理，在提取主要特征的时候，很多的噪声都被过滤掉了。跟下面将谈到的PCA的作用非常相关。

[](http://images.cnblogs.com/cnblogs_com/LeftNotEasy/201101/201101081455522470.png)

特征值的求法有很多，求一个D \* D的矩阵的时间复杂度是O(D^3), 也有一些求Top M的方法，比如说[power method](http://en.wikipedia.org/wiki/Power_method)，它的时间复杂度是O(D^2 \* M), 总体来说，求特征值是一个很费时间的操作，如果是单机环境下，是很局限的。

二. Fisherface 用于人脸识别

公式推理：

当得到[clip_image066](http://images.cnblogs.com/cnblogs_com/jerrylead/201104/201104212325074817.png)时，There’s one problem left to solve: The rank of S_{W}is at most (N-c), with Nsamples and cclasses. In pattern recognition problems the number of samples Nis almost always samller than the dimension of the input data (the number of pixels), so the scatter matrix S_{W}becomes singular (see [[RJ91]](http://docs.opencv.org/modules/contrib/doc/facerec/facerec_tutorial.html#rj91)). In [[BHK97]](http://docs.opencv.org/modules/contrib/doc/facerec/facerec_tutorial.html#bhk97) this was solved by performing a Principal Component Analysis on the data and projecting the samples into the (N-c)-dimensional space. A Linear Discriminant Analysis was then performed on the reduced data, because S_{W}isn’t singular anymore.

The optimization problem can then be rewritten as:

\begin{align*}
    W_{pca} & = & \operatorname{arg\,max}_{W} |W^T S_T W| \\
    W_{fld} & = & \operatorname{arg\,max}_{W} \frac{|W^T W_{pca}^T S_{B} W_{pca} W|}{|W^T W_{pca}^T S_{W} W_{pca} W|}
\end{align*}

The transformation matrix W, that projects a sample into the (c-1)-dimensional space is then given by:

W = W_{fld}^{T} W_{pca}^{T}