LDA-linear discriminant analysis

webdancerposted @ 2013年2月28日 20:41 in machine learningwith tags Machine learning , 7065 阅读

分类问题也可以用降维来理解，比如一个D D维的数据点x x，我们可以采用下面的映射进行线性的降维，

$$y = \theta^T x$$

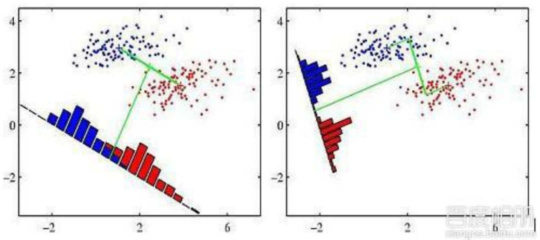
y=θTx

在计算出yy后，就可以选择一个阈值hh,来进行分类。正如我们在前面的PCA模型中看到的，降维会有信息的损失，可能会在降维过程中，丢失使数据可分的特征，导致分类的效果不理想。

那采用什么样的降维方式，可以尽量的在低维空间中保存原来数据在高维空间中的可分性（区分类别的特征）。一个常用的模型 linear discriminant analysis(LDA)就是用来做这个工作的，下面就具体的看一下LDA模型。

原理

LDA的基本原理就是最大化类间方差（between-class variance）和类内方差（within-class variance）的比率（注意这个variance用来理解，下面用到的定义实际上是variance的一个变形），使得降维后数据有最好的可分性。如果偷用软件工程里面用的术语的话，就是"高内聚，低耦合"，类内的数据内聚，方差小，而类间数据松散，方差大。通常来说，这要比只考虑类间的距离越大要好，如下图所示：



左边图只是考虑最大化每个类期望的最大距离，我们看到有很多点投影后重合了，丧失了标签信息；而右边是LDA投影，重合的点的数目减少了很多，能更好的保存标签信息。

模型

下面我们就来形式化这个过程，首先如何定义between-class variance和within-class variance？在Fisher提出的方法中，没有使用统计中标准的variance的定义，而是使用了一个称为scatter的概念，与variance时等价的，使用这个概念可能是为了后面的推导简洁。设数据集为$X = x_1, x_2, .., x_N$ X=x1,x2,..,xN，则scatter的定义为：

$$s = \sum_{n=1}^{N} (x_n - m)^T (x_n - m)$$

s=∑n=1N(xn−m)T(xn−m)

其中，$m = \frac{1}{N} \sum_{n=1}^{N} x_n$m=1N∑n=1Nxn。

类内方差很容易形式化，可以直接使用scatter来定义，然后把所有类别的scatter连加；那么类间的方差如何定义才能很好的让类之间的数据分的更开呢？当然应该有很多的数学关系很描述，在LDA中使用了下面这种方式，计算每个类别的期望，求期望之间的距离。先从简单的两类情况开始，然后拓展到多类的情况。

两类

设数据集合为$X = \{x_1, x_2, .., x_N\}$ X={x1,x2,..,xN},类别为C_1, C_2C1,C2,则这两类的数据期望为m_1, m_2m1,m2，计算公式如：

$$m_k = \frac{1}{N_k} \sum_{i \in C_k} x_i$$

mk=1Nk∑i∈Ckxi

$m_k$mk表示投影后的数据点的期望，则between-class variance的形式化定义为：

$$m_2 - m_1 = \theta^T (m_2 - m_1)$$

m2−m1=θT(m2−m1)

其中，$m_k = \theta^T m_k$mk=θTmk。within-class variance用within-scatter这个定义来表示，scatter是variance的变种（不用除以数据的数目），第C_kCk类的scatter定义为：

$$S_k^2 = \sum_{i \in C_k} (y_i - m_i)^2$$

Sk2=∑i∈Ck(yi−mi)2

其中，$y_i = \theta^T x_i$yi=θTxi。这样就可以得到目标函数：

$$J(\theta) = \frac{(m_2 - m_1)^2}{s_1^2 + s_2^2}$$

J(θ)=(m2−m1)2s12+s22

将上面的定义代入上式，可以得到式子：

$$\mathrm{maxarg}_\theta J(\theta) = \frac{\theta^T S_B \theta}{\theta^T S_W \theta}$$

maxargθJ(θ)=θTSBθθTSWθ

其中，S_B, S_WSB,SW分别称为between-class scatter和within-class scatter，表示如下：

$$S_B = (m_2 - m_1)(m_2 - m_1)^T; S_W = S_1 + S_2$$

SB=(m2−m1)(m2−m1)T;SW=S1+S2

其中,$S_k = \sum_{i \in C_k}(x_i - m_k)(x_i - m_k)^T$ Sk=∑i∈Ck(xi−mk)(xi−mk)T。下面要做的就是最优化目标函数$(x - m_k)$ (x−mk),对上面的式子求导数，让导数为0，则可以得到:

$$(\theta^T S_B \theta) S_W \theta = (\theta^T S_W \theta) S_B \theta$$
(θTSBθ)SWθ=(θTSWθ)SBθ

由于投影操作，我们只关心θθ的方向，上面的式子，可以去掉$(\theta^T S_B \theta), (\theta^T S_W \theta)$(θTSBθ),(θTSWθ),根据$S_B$ SB的定义，$S_B \theta$SBθ的方向与$(m_2 - m_1)$ (m2−m1)一致，我们可以得到:

$$\theta^* \propto S_W^{-1}(m_2 - m_1)$$
θ∗∝SW−1(m2−m1)

这个式子称为Fisher’s linear discriminant[1936],尽管这个式子不是一个判别式，只是选择了投影方向,不过只要我们选择一个阈值，然后就可以根据这个阈值进行分类了。(ps:使用求解generalized eigenvalue problem的方法求解导数为零的等式，也可以得到这个判别式)

多类

在多类问题时，将D D 维的向量x x 投影到$M < D$ M<D 维的y y ,投影矩阵方程为:

$$y = \Theta^T x$$
y=ΘTx

可以参照PCA文章中提到投影公式，这里ΘΘ是一个投影矩阵，每一个列向量表示一个投影方向Θ_kΘk。

设数据集合为$X = \{x_1, x_2, .., x_N\}$ X={x1,x2,..,xN},类别为$C_1, C_2, .., C_K$ C1,C2,..,CK。在多类的时候，过程与上面一样，不过由于between-class scatter 和within-class scatter不再是标量，需要更改一下我们需要优化的目标函数。首先看一下在原空间xx的定义，然后就可以类比到yy空间。

withinin-class scatter 与二类时的定义一样，如下表示:

$$S_W = \sum_{k=1}^{K} \sum_{i \in C_k} (x_i - m_k)(x_i - m_k)^T$$
SW=∑k=1K∑i∈Ck(xi−mk)(xi−mk)T

$m_k$mk定义与上面一致。

between-class scatter的定义，这里我们根据PRML里面论述的，首先定义一个S_T ST，然后根据$S_T = S_B + S_W$ ST=SB+SW，然后分解得到S_B SB。S_T ST的定义类似S_kSk，不过不在一个类别，而是在所有的数据集上进行计算。

$$\begin{aligned} S_T &= \sum_{n=1}^{N}(x_n - m)(x_n - m)^T \\ m &= \frac{1}{N}\sum_{n=1}^{N} x_n = \frac{1}{N}\sum_{k=1}^{K} N_k m_k \end{aligned}$$
ST=∑n=1N(xn−m)(xn−m)Tm=1N∑n=1Nxn=1N∑k=1KNkmk

所以得到:

$$\begin{aligned} S_B &= S_T - S_W \\ &= \sum_{n=1}^{N}(x_n - m)(x_n - m)^T - \sum_{k=1}^{K}\sum_{i \in C_k}(x_i - m_k)(x_i - m_k)^T \\ &= \sum_{k=1}^{K}\sum_{i \in C_k}(x_i - m)(x_i - m)^T - \sum_{k=1}^{K}\sum_{i \in C_k}(x_i - m_k)(x_i - m_k)^T \\ &= \sum_{k=1}^{K}\sum_{i \in C_k}\{(x_i - m)(x_i - m)^T - (x_i - m_k)(x_i - m_k)^T\} \\ &= \sum_{k=1}^{K}\{\sum_{i \in C_k} -x_i m^T + \sum_{i \in C_k} -m x_i^T + N_k m m^T + \sum_{i \in C_k} x_i m_k^T + \sum_{i \in C_k} m_k x_i^T - N_k m_k m_k^T\} \\ &= \sum_{k=1}^{K}\{-N_k m_k m^T - m N_k m_k + N_k m m^T + N_k m_k m_k^T + N_k m_k m_k - N_k m_k m_k^T\} \\ &= \sum_{k=1}^{K} N_k (m_k - m)(m_k - m)^T \end{aligned}$$
SB=ST−SW=∑n=1N(xn−m)(xn−m)T−∑k=1K∑i∈Ck(xi−mk)(xi−mk)T=∑k=1K∑i∈Ck(xi−m)(xi−m)T−∑k=1K∑i∈Ck(xi−mk)(xi−mk)T=∑k=1K∑i∈Ck{(x

这样我们就可以类比得到在投影空间的between-class scatter与within-class scatter:

$$\begin{aligned} \tilde{S}_W &= \sum_{k=1}^{K}\sum_{i \in C_k}(y_i - m_k)(y_i - m_k)^T \\ \tilde{S}_B &= S_T - S_W = \sum_{k=1}^{K} N_k (m_k - m)(m_k - m)^T \end{aligned}$$
S~W=∑k=1K∑i∈Ck(yi−mk)(yi−mk)TS~B=ST−SW=∑k=1KNk(mk−m)(mk−m)T

这样就可以得到目标函数，由于$\tilde{S}_W, \tilde{S}_B$ S~W,S~B不是标量，在目标函数中使用它们的行列式，

$$\text{maxarg}_\Theta J(\Theta) = \frac{|\tilde{S}_B|}{|\tilde{S}_W|}$$

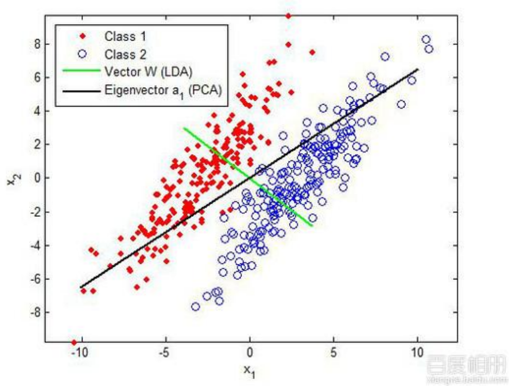
maxargΘJ(Θ)=|S~B||S~W|

类似在二类推到中的式子，可以得出：

$$\max\arg_{\Theta} J(\Theta) = \frac{|\tilde{S}_B|}{|\tilde{S}_W|} = \frac{|\Theta^T S_B \Theta|}{|\Theta^T S_W \Theta|}$$

maxargΘJ(Θ)=|S~B||S~W|=|ΘTSBΘ||ΘTSWΘ|

然后优化上面的函数(很直接，但是这里就不推导了，可能比较麻烦)，可以得出结论，投影矩阵由$S_W^{-1}S_B$ SW−1SB的特征最大特征向量决定，这样我们就可到了一个很简洁的公式， 与PCA不同的是，这里考虑到了类别信息,得到的投影方向对一些数据集合来说，会有很大不同，如下图：



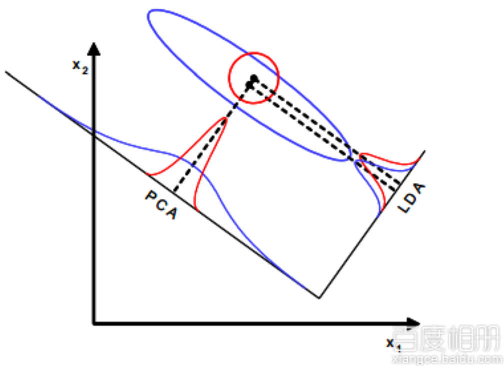
从上图中也可以看到，使用PCA投影后，数据在黑色的直线上基本不可分，而使用LDA投影，则可分性要比PCA好很多，这也说明了LDA在降维过程中保留了标签信息。

需要注意的地方是：

1. 由于S_B SB的秩最大为$K-1$ K−1,所以$S_W^{-1}S_B$ SW−1SB的特征向量数目不会超过$K-1$ K−1，所以我们投影后的$M <= (K-1)$ M<=(K−1)。
2. LDA也可以从normal class Density 通过最大似然估计得出。
3. $S_W^{-1}S_B$ SW−1SB中，用到了S_W SW的逆，但是S_W SW的最大秩为$N-K$ N−K，在很多计算中，特征数远大于样本数，使得S_W SW是奇异矩阵，所以这时候我们需要在LDA计算前，进行降维（采用PCA），使得S_W SW是非奇异的。

模型的局限性，主要体现在下面两个方面：

1. 根据上面的分析，LDA投影后最多只能保留$K-1$ K−1个特征，可能对一些问题来说，特征数目太少。
2. LDA本是参数估计方法，假设分布符合单峰的高斯分布，对于数据集合不符合的情况，没法保留标签信息。
3. 对那些由方差，而不是均值来区分的数据来说，LDA同样也没法处理，如下图所示:



应用

在人脸识别中，使用LDA降维，是一种常用的方法，形成的特征向量，称为fisher-face；此外，LDA也可以用在破产预测等方面。

引用：

[1]prml

[2]http://research.cs.tamu.edu/prism/lectures/pr/pr_l10.pdf

[3]http://www.intechopen.com/books/speech-technologies/nonlinear-dimensionality-reduction-methods-for-use-with-automatic-speech-recognition

[4]http://en.wikipedia.org/wiki/Linear_discriminant_analysis