leftnoteasy

关注于 机器学习、数据挖掘、并行计算、数学

机器学习中的数学(4)-线性判别分析（LDA），主成分分析(PCA)

版权声明：

本文由LeftNotEasy发布于http://leftnoteasy.cnblogs.com, 本文可以被全部的转载或者部分使用，但请注明出处，如果有问题，请联系wheeleast@gmail.com

前言：

第二篇的文章中谈到，和部门老大一宁出去outing的时候，他给了我相当多的机器学习的建议，里面涉及到很多的算法的意义、学习方法等等。一宁上次给我提到，如果学习分类算法，最好从线性的入手，线性分类器最简单的就是LDA，它可以看做是简化版的SVM，如果想理解SVM这种分类器，那理解LDA就是很有必要的了。

谈到LDA，就不得不谈谈PCA，PCA是一个和LDA非常相关的算法，从推导、求解、到算法最终的结果，都有着相当的相似。

本次的内容主要是以推导数学公式为主，都是从算法的物理意义出发，然后一步一步最终推导到最终的式子，LDA和PCA最终的表现都是解一个矩阵特征值的问题，但是理解了如何推导，才能更深刻的理解其中的含义。本次内容要求读者有一些基本的线性代数基础，比如说特征值、特征向量的概念，空间投影，点乘等的一些基本知识等。除此之外的其他公式、我都尽量讲得更简单清楚。

LDA：

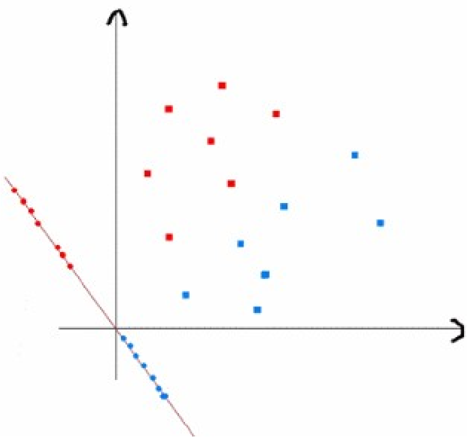
LDA的全称是Linear Discriminant Analysis（线性判别分析），**是一种supervised learning。**有些资料上也称为是Fisher's Linear Discriminant，因为它被Ronald Fisher发明自1936年，Discriminant这次词我个人的理解是，一个模型，不需要去通过概率的方法来训练、预测数据，比如说各种贝叶斯方法，就需要获取数据的先验、后验概率等等。LDA是在**目前机器学习、数据挖掘领域经典且热门**的一个算法，据我所知，百度的商务搜索部里面就用了不少这方面的算法。

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)：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：

$$y_k(x) = w_k^T x + w_{k0}$$

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：



最新评论

1. Re:机器学习中的数学(5)-强大的矩阵奇异值分解(SVD)及其应用

@echo_TJ当然不是。PCA需要求协方差矩阵，协方差矩阵就是方阵。...

--Stomach_ache

2. Re:机器学习中的算法(2)-支持向量机(SVM)基础

厉害，厉害，膜拜学习！！！

--miffymoon

3. Re:机器学习中的算法(2)-支持向量

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：

$$y = w^T x$$

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。

类别i的原始中心点为：（Di表示属于类别i的点)

$$m_i = \frac{1}{n_i}\sum_{x\in D_i} x$$

类别i投影后的中心点为：

$$\widetilde{m_i} = w^T m_i$$

衡量类别i投影后，类别点之间的分散程度（方差）为：

$$\widetilde{s_i} = \sum_{y\in Y_i}(y-\widetilde{m_i})^2$$

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：

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

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内的方差之和，方差越大表示一个类别内的点越分散，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。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.

$$S_i = \sum_{x\in D_i}(x-m_i)(x-m_i)^T$$

带入Si，将J(w)分母化为：

$$\widetilde{s_i} = \sum_{x\in D_i}(w^T x - w^T m_i)^2 = \sum_{x\in D_i} w^T(x-m_i)(x-m_i)^T w = w^T S_i w$$

$$\widetilde{s_1}^2 + \widetilde{s_2}^2 = w^T(S_1+S_2)w = w^T S_w w$$

同样的将J(w)分子化为：

$$|\widetilde{m_1}-\widetilde{m_2}|^2 = w^T(m_1-m_2)(m_1-m_2)^T w = w^T S_B w$$

这样损失函数可以化成下面的形式：

$$J(w) = \frac{w^T S_B w}{w^T S_w w}$$

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：

$$c(w) = w^T S_B w - \lambda(w^T S_w w - 1)$$

$$\Rightarrow \frac{dc}{dw} = 2S_B w - 2\lambda S_w w = 0$$

机(SVM)基础

x为二维向量，对应的分离超平面是w1*x1+w2*x2+b=0，是一条直线。如果你预测的值(x11,x12)带入超平面方程后，w1*x11+w2*x12+b>0，则预测的值是+1类；如果w1*x11+......

--yeepom

4. Re:机器学习中的数学(5)-强大的矩阵奇异值分解(SVD)及其应用

博主，PCA中利用特征值进行降维时对象只局限于方阵？？？

--echo_TJ

5. Re:机器学习中的数学(1)-回归(regression)、梯度下降(gradient descent)

谢谢楼主的分享，之前看过stanford的视频，讲解的很清楚。求一份prml的书，邮箱mtxing69@163.com，感谢。

--mtx

阅读排行榜

1. 机器学习中的数学(5)-强大的矩阵奇异值分解(SVD)及其应用(135796)
2. 机器学习中的算法(2)-支持向量机(SVM)基础(98838)
3. 机器学习中的算法(1)-决策树模型组合之随机森林与GBDT(87145)
4. 机器学习中的数学(4)-线性判别分析（LDA），主成分分析(PCA)(86194)
5. 机器学习中的数学(1)-回归(regression)、梯度下降(gradient descent)(77766)

评论排行榜

1. 机器学习中的数学(5)-强大的矩阵奇异值分解(SVD)及其应用(60)
2. 机器学习中的算法(2)-支持向量机(SVM)基础(37)
3. 机器学习中的数学(1)-回归(regression)、梯度下降(gradient descent)(34)
4. 机器学习中的数学(4)-线性判别分析（LDA），主成分分析(PCA)(27)
5. 机器学习中的数学(2)-线性回归，偏差、方差权衡(23)

推荐排行榜

1. 机器学习中的数学(5)-强大的矩阵奇异值分解(SVD)及其应用(65)
2. 机器学习中的算法(2)-支持向量机(SVM)基础(35)
3. 机器学习中的数学(1)-回归(regression)、梯度下降(gradient descent)(27)
4. 机器学习中的算法(1)-决策树模型组合之随机森林与GBDT(26)
5. 为什么Hadoop将一定会是分布式计算的未来？(23)

$$aw$$

$$\Rightarrow S_B w = \lambda S_w w$$

这样的式子就是一个求特征值的问题了。

对于N(N>2)分类的问题，我就直接写出下面的结论了：

$$S_W = \sum_{i=1}^{c} S_i$$

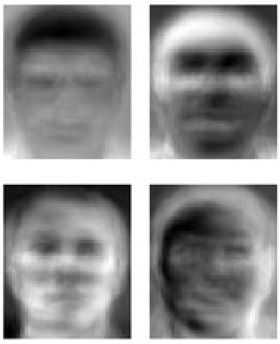
$$S_B = \sum_{i=1}^{c} n_i (m_i - m)(m_i - m)^T$$

$$S_B w_i = \lambda S_w w_i$$

这同样是一个求特征值的问题，我们求出的第i大的特征向量，就是对应的Wi了。

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。



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，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。

PCA：

主成分分析（PCA）与LDA有着非常近似的意思，LDA的输入数据是带标签的，而PCA的输入数据是不带标签的，所以PCA是一种unsupervised learning。LDA通常来说是作为一个独立的算法存在，给定了训练数据后，将会得到一系列的判别函数（discriminate function），之后对于新的输入，就可以进行预测了。而PCA更像是一个预处理的方法，它可以将原本的数据降低维度，而使得降低了维度的数据之间的方差最大（也可以说投影误差最小，具体在之后的推导里面会谈到）。

方差这个东西是个很有趣的，有些时候我们会考虑减少方差（比如说训练模型的时候，我们会考虑到方差-偏差的均衡），有的时候我们会尽量的增大方差。方差就像是一种信仰（强哥的话），不一定会有很严密的证明，从实践来说，通过尽量增大投影方差的PCA算法，确实可以提高我们的算法质量。

说了这么多，推推公式可以帮助我们理解。**我下面将用两种思路来推导出一个同样的表达式。首先是最大化投影后的方差，其次是最小化投影后的损失（投影产生的损失最小）。**

最大化方差法：

假设我们还是将一个空间中的点投影到一个向量中去。首先，给出原空间的中心点：

$$\bar{x} = \frac{1}{N} \sum_{n=1}^{N} x_n$$

假设u1为投影向量，投影之后的方差为：

$$\frac{1}{N} \sum_{n=1}^{N} \{u_1^T x_n - u_1^T \bar{x}\}^2 = u_1^T S u_1$$

上面这个式子如果看懂了之前推导LDA的过程，应该比较容易理解，如果线性代数里面的内容忘记了，可以再温习一下，优化上式等号右边的内容，还是用拉格朗日乘子法：

$$u_1^T S u_1 + \lambda_1 (1 - u_1^T u_1)$$

将上式求导，使之为0，得到：

$$S u_1 = \lambda_1 u_1$$

这是一个标准的特征值表达式了，λ对应的特征值，u对应的特征向量。上式的左边取得最大值的条件就是λ1最大，也就是取得最大的特征值的时候。假设我们是要将一个D维的数据空间投影到M维的数据空间中（M < D），那我们取前M个特征向量构成的投影矩阵就是能够使得方差最大的矩阵了。

最小化损失法：

假设输入数据x是在D维空间中的点，那么，我们可以用D个正交的D维向量去完全的表示这个空间（这个空间中所有的向量都可以用这D个向量的线性组合得到）。在D维空间中，有无穷多种可能找这D个正交的D维向量，哪个组合是最合适的呢？

假设我们已经找到了这D个向量，可以得到：

$$x_n = \sum_{i=1}^{D} \alpha_{ni} u_i$$

我们可以用近似法来表示投影后的点：

$$\tilde{x}_n = \sum_{i=1}^{M} z_{ni} u_i + \sum_{i=M+1}^{D} b_i u_i$$

上式表示，得到的新的x是由前M 个基的线性组合加上后D - M个基的线性组合，注意这里的z是对于每个x都不同的，而b对于每个x是相同的，这样我们就可以用M个数来表示空间中的一个点，也就是使得数据降维了。但是这样降维后的数据，必然会产生一些扭曲，我们用J描述这种扭曲，我们的目标是，使得J最小：

$$J = \frac{1}{N} \sum_{n=1}^{N} || x_n - \tilde{x}_n ||^2$$

上式的意思很直观，就是对于每一个点，将降维后的点与原始的点之间的距离的平方和加起来，求平均值，我们就要使得这个平均值最小。我们令：

$$\frac{\partial J}{\partial z_{nj}} = 0 \Rightarrow z_{nj} = x_n^T u_j, \quad \frac{\partial J}{\partial b_j} = 0 \Rightarrow b_j = \bar{x}^T u_j$$

将上面得到的z与b带入降维的表达式：

$$x_n - \tilde{x}_n = \sum_{i=M+1}^{D} \{(x_n - \bar{x}) u_i\} u_i$$

将上式带入J的表达式得到：

$$J = \frac{1}{N} \sum_{n=1}^{N} \sum_{i=M+1}^{D} (x_n^T u_i - \bar{x}^T u_i)^2 = \sum_{i=M+1}^{D} u_i^T S u_i$$

再用上拉普拉斯乘子法（此处略），可以得到，取得我们想要的投影基的表达式为：

$$S u_i = \lambda_i u_i$$

这里又是一个特征值的表达式，我们想要的前M个向量其实就是这里最大的M个特征值所对应的特征向量。证明这个还可以看看，我们J可以化为：

$$J = \sum_{i=M+1}^{D} \lambda_i$$

也就是当误差J是由最小的D - M个特征值组成的时候，J取得最小值。跟上面的意思相同。

下图是PCA的投影的一个表示，黑色的点是原始的点，带箭头的虚线是投影的向量，Pc1表示特征值最大的特征向量，pc2表示特征值次大的特征向量，两者是彼此正交的，因为这原本是一个2维的空间，所以最多有两个投影的向量，如果空间维度更高，则投影的向量会更多。

总结：

本次主要讲了两种方法，PCA与LDA，两者的思想和计算方法非常类似，但是一个是作为独立的算法存在，另一个更多的用于数据的预处理的工作。另外对于PCA和LDA还有核方法，本次的篇幅比较大了，先不说了，以后有时间再谈：

参考资料：

prml bishop，introduce to LDA（对不起，这个真没有查到出处）

分类: 机器学习, 数学

标签: machine learning, 机器学习, 线性分类, 主成分分析, LDA, PCA

« 上一篇：机器学习中的数学(3)-模型组合(Model Combining)之Boosting与Gradient Boosting
» 下一篇：机器学习中的数学(5)-强大的矩阵奇异值分解(SVD)及其应用

评论

#1楼[楼主] 2011-01-08 23:14 LeftNotEasy

@ 丕子
好的：)

支持(0) 反对(0)

#2楼 2011-01-10 11:16 Draco9

支持一下~

P.S. 有一个小错误。
"分母表示每一个类别内的方差之和，方差越大表示一个类别内的点越分散，分子为两个类别各自的中心点的距离的平方，我们最小化J(w)就可以求出最优的w了。"

应该是最大化J(w)。

支持(0) 反对(0)

#3楼[楼主] 2011-01-10 12:59 LeftNotEasy

@ Draco9
这里确实写错了，马上改正过来。谢谢关注：)

支持(0) 反对(0)

#4楼 2011-01-13 14:37 Leo Zhang

LDA让我想到了projection pursuit method 当时用GA去求最佳投影方向 结果被导师批了

支持(0) 反对(0)

#5楼[楼主] 2011-01-13 16:55 LeftNotEasy

@ Leo Zhang
GA是什么？愿求其详~

支持(0) 反对(0)

#6楼 2011-01-13 22:08 Leo Zhang

@ LeftNotEasy
就是遗传算法了 不过设计的不好会导致收敛很慢

支持(0) 反对(0)

#7楼[楼主] 2011-01-13 22:19 LeftNotEasy

@ Leo Zhang
求投影方向为什么要用GA去求呢，LDA这种多项式复杂度的不是更适合吗

支持(0) 反对(1)

#8楼 **2011-01-14 12:34** **Leo Zhang**

@ LeftNotEasy
对滴 所以呢当时就被批了

支持(0) 反对(0)

#9楼 **2011-01-14 17:52** **夏日疯狂**

用拉格朗日法是可以求极值的，但是你怎么确定是极大值还是极小值？没有见说明？是求出以后再比较得到最大值？

支持(0) 反对(0)

#10楼[楼主] **2011-01-15 17:41** **LeftNotEasy**

@ 夏日疯狂
应该这样看，对于分子和分母都一定是恒为正的，而且拉格朗日乘子式求导有下面的式子：Sb * w = lambda * Sw * w，将其带入J(w)得到，当w满足等式的时候，J(w) = lambda, 而当w为第一个eigen vector的时候，lambda取得第一个也是最大的eigen value， 对第二、第三个eigen value以此类推

支持(0) 反对(0)

#11楼 **2011-02-12 10:48** **行隐**

收益匪浅，之前看到1楼的转载，发现3楼发现的问题，回来得到验证，感谢博主的分享

支持(0) 反对(0)

#12楼 **2011-02-12 19:36** **emy_yu**

总结的很详细，之前学SVM一直很模糊，谢谢~~

支持(0) 反对(0)

#13楼[楼主] **2011-02-24 17:46** **LeftNotEasy**

@ 行隐
谢谢关注：）

支持(0) 反对(0)

#14楼 **2011-04-10 22:53** **JerryLead**

顶，好文，学习了。
p.s.有个地方没看明白：
“ 衡量类别i投影后，类别点之间的分散程度（方差）为：”和
“带入Si，将J(w)分母化为：”下面的Si的公式的Si是不是应该是Si的平方啊

支持(0) 反对(0)

#15楼[楼主] **2011-04-12 11:03** **LeftNotEasy**

@ JerryLead
我回头看了一下，应该是你说的这样，文章中的有一点问题，不过是图片，改起来有一点小麻烦，呵呵，我之后有时间再改改，谢谢关注：）

支持(0) 反对(1)

#16楼 **2012-03-06 12:43** **会飞的瓶子**

谢谢您的分享，对我很有帮助！

支持(0) 反对(0)

#17楼 **2012-06-20 22:47** **幻想时代5**

$$y_k(x) = w_k^T x + w_{k0}$$

楼主，我弱弱问下这个Wk是怎么个形式？用来干什么的？

支持(0) 反对(0)

#18楼 **2012-07-31 15:26** **ello**

> 引用
> 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：

支持(0) 反对(0)

#19楼 **2012-07-31 15:28** **ello**

损失函数不是应该越小越好么？ 为什么“最大化J(w)就可以求出最优的w了” 是不是应该把这个损失函数的分子分母倒过来？

支持(0) 反对(0)

#20楼 **2012-11-07 10:11** **silence1214**

在线性分类器中中，说道对于所有的j吗如果
YK>yJ 那么这些模式就属于类K，这个地方我觉得如果LZ写成yk(x)>yj(x)就好多了，这个地方最少让我一下子没明白过来，还是复习了线性判别函数才回头来看明白

支持(0) 反对(0)

#21楼 **2012-11-07 14:41** **silence1214**

@ ello这个损失函数叫法有错误，应该叫做目标函数

支持(0) 反对(0)

#22楼 **2013-10-20 09:30** **thinkercui**

楼主有问题的地方尽量改正吧。当大家发现比较明显的问题时，就不敢相信文章的权威性了。。。

支持(0) 反对(0)

#23楼 2014-04-04 10:58 郭志通

感觉不是很直观，投影后的损失最小，损失的是什么？怎么样才算损失最小？

支持(0) 反对(0)

#24楼 2014-12-27 10:40 zhaoyang10

你的公式里，方差的右上角应该都加上^2的符号。有几个地方写错了，下面的公式推导不出来。

支持(0) 反对(0)

#25楼 2015-03-28 11:20 迁客在行动

二分类问题中的最后一个结论
为什么说这个是求特征值的问题？SB和Sw都是方阵，W是一个列向量。而且又不能保证Sw是可逆矩阵，应该说在源数据特征维数超过2时，Sw的绝对不是特征矩阵，Sw = S1 + S2， S1和S2都是由一个列向量乘以行向量得到的，所以秩不可能超过1，而Sw就不可能超过2，，因为矩阵的和的秩小于等于两个矩阵的秩的和。

支持(0) 反对(1)

#26楼 2015-04-08 17:05 羊羊1995

对于才入门的我，看到这样比较通俗易懂的文章，很是开心，楼主棒棒哒

支持(0) 反对(0)

#27楼 2015-10-20 22:22 IAmHere

@ 迁客在行动
S1 S2是由多组列向量乘行向量得到的矩阵求和

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 登录 或 注册，访问网站首页。

最新IT新闻:
· Windows 10强制升级致用户损失300万元大单 专家建议提起诉讼
· 美国首次发现能"对抗"抗生素"最后一道防线"的细菌
· 上海环球港开业：华东首家采用新设计的店
· 美国宇航局发现火星存在冰河时代
· 芬兰政府指责微软对诺基亚始乱终弃 承诺一个都没兑现
» 更多新闻...

最新知识库文章:
· 程序员的黄金时代
· 快速学习者的高效学习策略
· 一个前端的自我修养
· 架构漫谈（九）：理清技术、业务和架构的关系
· 架构漫谈（八）：从架构的角度看如何写好代码
» 更多知识库文章...