Olivetti人脸数据的判别

王政喻

2014年6月

摘要

使用传统的 $Fihser$ 线性判别函数对于 Olivetti Faces Dataset 做分类，可以得到相当不错的结果。然而，美中不足的地方是 $Fisher$ 线性判别函数错误地将第 5 个人分到第 40 个人去。本文尝试使用向量支持机 (Support Vector Machines) 的判别方法，对该数据重新做判别分析，期望改善 $Fisher$ 线性判别函数的缺失。其中我们引入判别误差项以及内积的概念，并给予误差项适当的权重。我们研究在不同的权重与不同的内积定义下，所得出的判别结果有何不同。最后从其中选择一个相对较优的 SVM ，取代原来的 $Fisher$ 线性判别函数，提高判别的精确度。

关键词: Olivetti Faces Dataset、$Fisher$ 线性判别、向量支持机、SVM

1 引言

随着统计工具以及数学理论的发展，人脸识别领域在近几十年来有重大的突破。识别人脸有许多方法，其中一种是透过对人脸数据进行旋转，使得该脸图与其他脸图有所区分，进而达到识别人脸的目的。AT&T 剑桥实验室在 1992 年 4 月到 1994 年 4 月，以不同的限制条件 (包括光照、时间点、脸部表情等)，记录了 40 个人的脸图，该数据为 Olivetti Faces Dataset (`http://cs.nyu.edu/~roweis/data.html`)。过去有诸多文章以 Olivetti Faces Dataset 进行判别 (discrimination) 分析，并且都得到了不错的结果。笔者发现 $Fisher$ 线性判别量虽然能以近乎完美的判别机制来区分这 40 个人，但仍然有极小的误差率。因此笔者尝试使用其他方法进行判别，希望提高判别的准确度。底下我们先演示原始的 $Fisher$ 线性判别。

2 初步判别分析

AT&T 剑桥实验室对每个人搜集了 10 张脸图，解析度为 64×64，因此数据结构为 400×4096。由于变量过多，我们使用主成分分析 (Principal Component Analysis, 简称 PCA) 进行降维后，再对原数据的得分做 $Fisher$ 线性判别。

2.1 PCA 降维

首先我们挑出每个人的最后一张脸图作为检验 (test) 集，其余 9 张为训练 (train) 集，并将两组数据标准化。对训练集做主成分分析后，我们发现前 80 个主成分便能解释 92% 左右的变异，因此我们将数据结构降成 360×80 以利分析。由图 1，我们可以看到该数据用主成分降维后，在一定程度上将每个人分开来，显示主成分降维有效。同样地，我们也对测试集做 PCA，得到一个 40×80 的数据阵。

现在我们对降维后的训练集做 $Fisher$ 线性判别分析，得到一个线性判别量，并以此判别量对测试集进行判别，得到如下结果:

原组别	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
分类后	1	2	3	4	40	6	7	8	9	10	11	12	13	14	15
原组别	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
分类后	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
原组别	31	32	33	34	35	36	37	38	39	40					
分类后	31	32	33	34	35	36	37	38	39	40					

表 1: 检验集分类结果

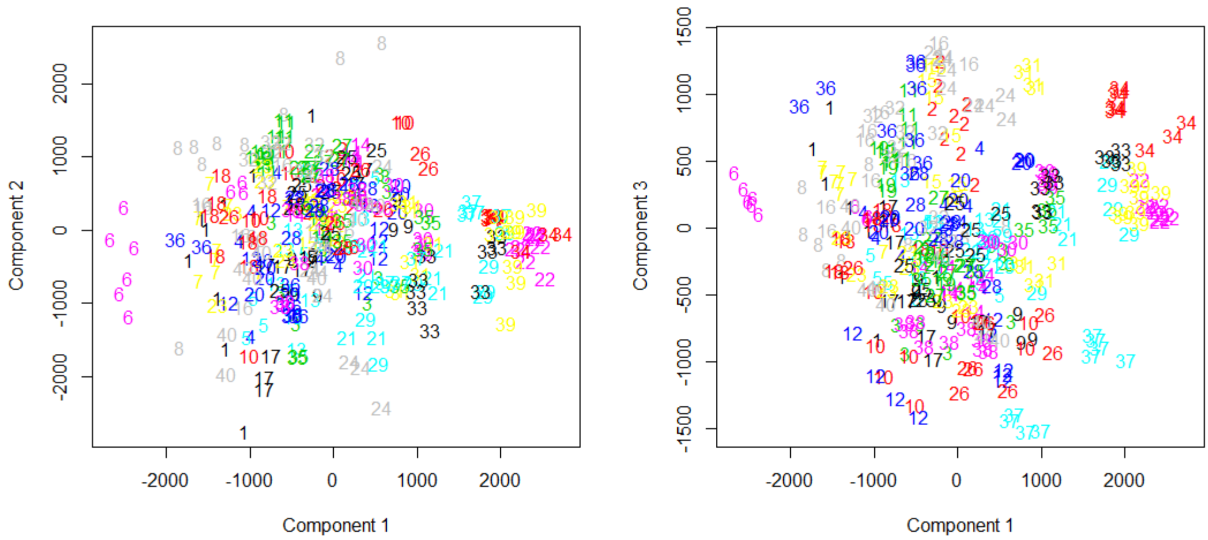

图 1: 主成分得分

表 1 显示 *Fisher* 线性判别结果近乎完美，仅出现一个小错误，即第 5 个人被分到第 40 个人去。这样的结果令我们感到好奇，为何只有这二个人判别错误？我们尝试将第 5 个人与第 40 个人挑选出来，做聚类分析。如果二者轮廓相似的话，那么聚类结果应该会把二者混在一起，不容易将树状图区分为二个群体。

图 2: 第 5 个人与第 40 个人脸图

2.2 聚类分析

图 2 分别画出第 5 与第 40 人的 10 张脸图，由该图我们可以直观认为两者的轮廓确实有些相似，因而造成判别上的错误。底下我们利用实际的数据，计算二者的相对距离

(相异度)，并以 Average Linkage 做层次聚类分析，结果如图 3 所示:

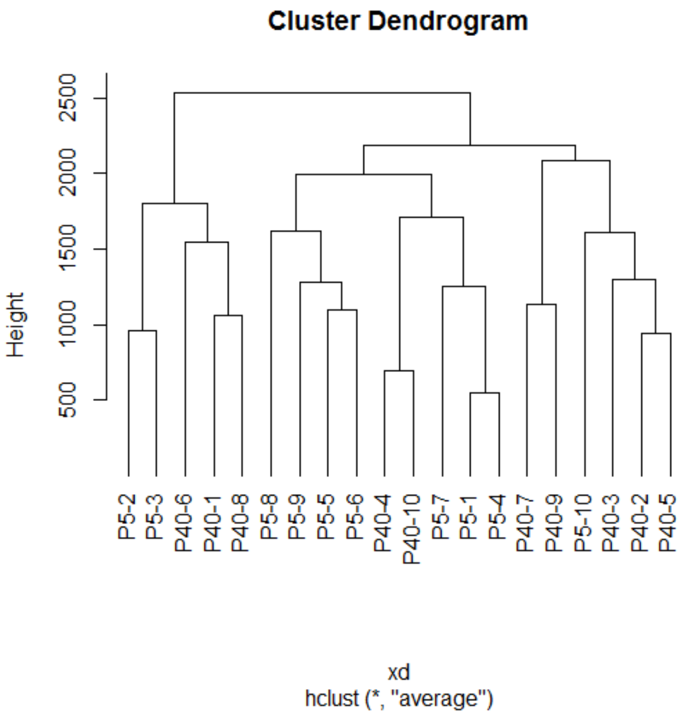

图 3: 聚类结果

在图3的树状图中，P5-2 表示第 5 个人的第 2 张脸图，其余依此类推。根据树状图，我们可以看出这 20 张脸图混合在一起，难以将树状图区分为二类。换句话说，我们无法将每个人的 10 张图近似地聚在一起，归类为同一类。这样的混合结果与 2.1 节的判别结果相一致，说明二者间的相似度造成了微小的判别失误。

现在，我们改用向量支持机 (Support Vector Machines, 简称 SVMs) 判别方法，重新对该数据进行分析，期望能将全部的人予以分开。同样的，我们使用主成分降维后的数据来做判别。

3 SVMs

SVMs 可以用来对数据做分类，其原理是将原始数据映射到某个坐标系上，该坐标系的维度可以与原始数据相同，也可以与原始数据不同。由于坐标变换有非常多的选择，我们只考虑能将数据最大程度分离的坐标系。透过这样的选择，在新的坐标系上，我们就可以尽可能的将二个类分开，达到分类的目的，进而得到一个判别机制。SVMs 的形式种类非常之多，我们经过多方尝试后，只挑选其中能将 Olivetti Faces Dataset 分离的 SVM，而其他种类的详细介绍请见`http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf`。

3.1 带松弛变量 (slack variable) SVM

很多时后我们无法找到能完美分离原始数据的坐标系，通常只能近似地分离。因此我们引入一个误差项，即松弛变量，定义该误差项为错误分类的数据点到分割线的距离。我们的目标除了最大程度上分离数据以外，还必须考虑误差项的大小，由此得出的判别量就是我们想要的结果。我们常常给松弛变量一个权数 C ，因此该模型也称为 C-SVM 。在求解的过程中，为了方便计算，我们必须考虑内积的形式。底下我们以多项式内积的方法进行求解，并分别演示 2 阶与 1 阶的结果。

3.2 判别结果

2 阶多项式内积

我们用 C-SVM 进行判别，并且在 2 阶多项式内积的基础下得到表 2 的结果。表 2 列出在不同的 C 值下，2 阶多项式内积 C-SVM 的判别误差率。我们可以看到该判别函数对于训练集的判别误差率为 0 ，这是应当的结果，因为该判别函数就是由训练集所制造出来的。另一方面，我们用这个判别函数对测试集做判别，得出错误率 10% 的结果。而原来的 *Fisher* 判别量只对第 5 个人判别错误，失误率为 1/40=2.5% 。这样的结果固然令人失望，但也暗示着该数据在高维空间的分类效果并不显著。因此我们将内积形式改为 1 阶，尝试能否用线性的方法将数据集分类。

	C	train	test
1	0.06	0.00	0.10
2	0.12	0.00	0.10
3	0.25	0.00	0.10
4	0.50	0.00	0.10
5	1.00	0.00	0.10
6	2.00	0.00	0.10
7	4.00	0.00	0.10
8	8.00	0.00	0.10
9	16.00	0.00	0.10

表 2: 2 阶多项式内积判别误差率

	C	train	test
1	0.06	0.00	0.00
2	0.12	0.00	0.00
3	0.25	0.00	0.00
4	0.50	0.00	0.00
5	1.00	0.00	0.00
6	2.00	0.00	0.00
7	4.00	0.00	0.00
8	8.00	0.00	0.00
9	16.00	0.00	0.00

表 3: 1 阶多项式内积判别误差率

1 阶多项式内积

此处所有的计算方法与2阶多项式内积如出一轍，只是内积的形式改成 1 阶，表 3 列出所有的判别结果。我们发现，将 C-SVM 的内积形式改成 1 阶以后，该判别函数呈现出完美的判别结果，这就是我们最终想要的判别函数。

4 结论

经过实验分析，我们知道由 Olivetti Faces Dataset 训练集制造出来的 $Fisher$ 线性判别量，无法完美的将第 5 个人与第 40 个人分开。在检视这两个人的分层树状图后，也得出二人脸图混合的结果。于是我们改用向量支持机来做判别，发现在内积为 2 阶多项式的 C-SVM 下，错分的频率反而提高；而当内积改为 1 阶后，C-SVM 则表现得相当完美。因此内积选定 1 阶多项式的 C-SVM ，对于该数据的分类效果最佳。

一般线性 C-SVM 的内积数学式为 $< x, y >$ ，而非线性的 d 阶多项式内积为 $(< x, y > +c)^d$ ，其中 $c, d \in C$ 。当我们选定 1 阶多项式时，实际上就是将线性内积加上一个常数 c 。根据 1 阶 C-SVM 的判别结果显示该数据是线性可分的。然而，同为线性的 $Fisher$ 判别量却无法完美的将数据切割开来，推测可能的原因有二。其一为 C-SVM 考虑了误差项，并且给予权重 C 。有了误差项，在建构判别量的过程中就必须考量错分的频率，而不单单只考虑分类数据的间隔。其二，内积加入了常数项 c ，使得距离可以做适当的调整，并藉此影响最后的判别函数。本文仅描述实验的结果，详细的数学推导则需要进一步的研究，或参考其他论文。

参考文献

[1] Chih-Chung Chang, and Chih-Jen Lin (2001), LIBSVM: A Library for Support Vector Machines

[2] Dr.Jassim T.Sarsouh, and Dr.Kadhem M.Hashem (2007), Clustering of Human Face images with different rotation angles, Journal of University of Thi-Qar, Vol.3, No.1

[3] Bien, J., and Tibshirani, R. (2011), Hierarchical Clustering with Prototypes via Minimax Linkage, The Journal of the American Statistical Association

[4] Delbert Dueck (2009), Affinity Propagation: Clustering Data by Passing Messages

[5] Alexandros Karatzoglou, David Meyer, and Kurt Hornik (2006), Support Vector Machines in R, Journal of Statistical Software, Vol.15, Issue 9