基于线性分类器的手写数字识别

黄旻浩
(香港中文大学,中国香港特别行政区 999077)

摘 要:线性分类器作为理解最简单表现最直观的算法之一,在众多更新更复杂的算法的涌现之后,依然在模式识别的应用中有一席之地,有被学习的必要。本文首先建立了一个完整的线性分类器进行手写数字识别,使用 MATLAB 的研究环境和 MNIST 的手写数据库样本。首先对于待识别的样本进行预处理,建立线性分类器,使用样本集进行训练并分类,再使用测试集得到其分类效果的数据。为了不同模式识别样本的性能,本文选取了 K 均值聚类,BP 神经网络和 SVM 算法,分别建立了分类器后,使用相同的样本集进行训练并测试其性能,从识别速度和准确性进行比较。最后本文对不同算法的测试效果进行比较,总结,分析各个识别算法的优劣。建立用户界面直观反映各个分类器的优劣和使用效果。

关键词:模式识别;线性分类器;聚类分析;BP 神经网络;SVM 算法
中图分类号:TP391.41 **文献标识码**:A **文章编号**:2096-4390(2019)33-0058-02

1 研究背景

模式识别就是使得计算机能够做到本来只有真人能去完成的任务,使其拥有人之前独有的对于各种事物进行接受信息,分析信息,描述事物,和自主判断的能力。事实上,模式识别是我们产生自我认识和对世界产生印象的第一步。人类在日常生活中在获取信息,处理信息和输出信息时,都在作何识别,描述,分类,再处理的工作,也就是说人脑会不断地进行着模式识别。而对于人类而言最基础的活动模式识别,对于没有模糊判断能力的机器而言却充满了难度。

所以,使得机器具备模式识别的能力,就像教导人类儿童学会识别形状和数字,对于社会生产力的发展和人工智能的进步有着重要的意义,也是实现接下来的模式识别更复杂的应用的初步实践和经验累积。

2 图片的预处理

对于训练样本和测试样本,我们对于获取的手写数字图片录入到计算机后,提取其灰度矩阵,并选用中值滤波进行去噪,基于 OTSU 算法的二值化,确定边界后切割并最终再归一化,得到所需要的 01 数值矩阵。

对于获得的数值矩阵的特征提取,我们网格统计提取法,鉴于 MNIST 数据库的数据格式,我们选用 4*4 大小的网格,获取 49 行的特征矩阵。

3 线性分类器的原理和建立

3.1 线性分类器的原理

当样本通过变换映射为特征向量以后,它就成为了特征空间中的点。而由于每个类中的样本会具有某些共性,即特征会有不同,那么属于一个类的样本集的点集,总是会与别的类的点集相分离,那么如果我们可以找到一个函数,能够把不同的点集相分离,那我们的任务也就解决了。由于判别函数法不依赖于概率密度分布的统计学知识,我们可以理解为将样本通过他们的特征用几何方法,将整个空间分解为不同类的子空间。

判别函数法可以根据边界所代表的函数划分为线性和非线性分类器。由于线性分类器涉及数学方法较为简单,实现更简便,我们选取了线性分类器作为本文研究方向。

对于手写数字识别,我们可以知道一共有 10 类模式 $t_0, t_1 \cdots\cdots t_9$,而我们预处理后,共有 49 个特征值,那么我们可以用 $X=(x_1, \cdots\cdots x_{49})^T$ 来表示样本。由于有 10 个模式类,那么线性判别函数形式为:

$$d(X)=w_1x_1+w_2x_2+\cdots\cdots+w_nx_n+w_{n+1}=W_0^T+w_{n+1}$$

由于有 10 个模式类我们就需要给出 10 个判别函数:$d_0(X)$,$d_1(X)$,……$d_9(X)$,若 X 属于第 i 类,则有:

$$d_i(X)>d_j(X),(j=0,\cdots\cdots 9,\ j\neq i)$$

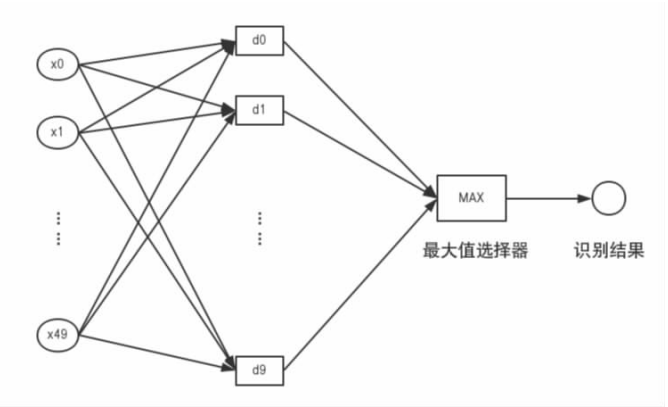

图 1 线性多类分类器形式

3.2“奖惩”算法

我们使用判别函数最大值的方式,由于手写数字识别的 10 个类,我们需要 10 个函数。

若在第 k 次迭代时,样品 X(k)属于 t_i 类,在“奖惩算法”中,我们需要对 10 个函数都进行计算。

3.2.1 若 $d_i[X(k)]>d_j[X(k)]$则权矢量不需要加以修正。

$$\begin{cases}W_i(k+1)=W_i(k)\\W_j(k+1)=W_j(k)\end{cases}$$

3.2.2 若 $d_i[X(k)]\leq d_j[X(k)]$则按下式进行修正:

$$\begin{cases}W_i(k+1)=W_i(k)+CX(k)\\W_j(k+1)=W_j(k)-CX(k)\end{cases}$$

3.3 测试结果

以 10000 个样本作为测试集,进行测试,得到数据如下:

作者简介:黄旻浩(1996-),男,汉族 江苏省无锡市人,香港中文大学研究生在读,理学硕士在读,统计机器学习。

样本数	1000	5000	10000	20000	30000	40000
正确率	72.25%	78.38%	84.01%	84.65%	83.41%	83.66%
时间	0.404s	0.621s	0.748s	1.172s	1.439s	1.947s

我们可以看到,在训练集样本容量增加超过 10000 后,正确率可达到 84%上下波动,在测试了 25000 和 35000 个样本后,我们发现在训练集容量为 30000, 达到极大值 82.41%。随着样本集容量增加,训练时间也会相应增加,每增加 10000 个样本,训练时间增加量会递增,但总体在 0.5s 以内,扔在可控范围之内。虽然得到了能够接受的训练成果,而且运算速度较快,算法实现简单, 但是这样的正确率显然是不足以投入实际应用中去的。

4 其他算法的原理概述及建立

为了更好地了解模式分析和线性分类器在手写数字识别上的效果,本文会继续对其他主要模式识别算法进行建立并比较结果。

4.1 聚类分析

聚类分析是在不知道一批样品中样品的类别时, 直接根据一定的算法,将特征相近的类归为一类。本文选用 K 均值动态聚类方法。

K 均值测试结果:

以 10000 个样本作为测试集,得到数据如下:

样本数	1000	5000	10000	20000	30000	40000
正确率	58.05%	56.30%	57.92%	57.29%	55.97%	56.81%
时间	0.506s	1.182s	2.348s	6.787s	11.334s	16.737s

用 K 均值进行聚类得到的结果均仅在 57%左右,样本集的增加并不会有效提升识别正确率,在低样本容量时,正确率反而会提升,这是由于数字样本容量大,特征值多,几何空间具有复杂性,而 K 均值算法仅仅是通过距离定义,在对于相似的数字处理时受到手写变形的影响过大。增加样本集会很大地增加分类时间,每增加 10000 个样本,会增加 5s 左右,但是可以发现,在样本很少的时候,聚类分析依然可以工作。但是由于正确率和输出结果不近如人意,可认为我们在进行手写数字识别的时候,可以不考虑使用 K 均值的方法直接聚类分类。

4.2 BP 神经网络

神经网络试图模拟推理和自主学习, 使计算机更接近人脑的自组织和并行处理功能。神经网络就是一个从输入点到输出点的非线性映射。它的学习方式就是在输入样本的过程中不断地改进参数和阈值来实现学习,实现模式分类,神经网络由其对数据分布无相关要求的特点,受到了广泛的应用。

BP 神经网络测试结果:

进行测试,得到数据如下:

样本数	1000	5000	10000	20000	30000	40000
正确率	71.15%	81.24%	82.87%	82.17%	83.00%	83.13%
时间	72.06s	98.207s	84.446s	74.452s	67.675s	69.514s

我们可以发现,BP 神经网络在训练集在 10000 以内时,训练速度较快, 且在训练集达到 5000 以上时, 正确率在能达到 80%以上,随着样本集的增多,正确率无明显上升,但训练时间明显上升,但运算时间会下降。我们可以认为,对于需要多次使用的识别,在我们训练集足够准确,调试够准确,允许误差设置更小,BP 网络是很好的选择。

4.3 支持向量机

支持向量机算法的最初研究方法是针对两类线性可分问题,其原理就是在两类样本中确定一个超平面,将两类样本分开,并使其具有最大间隔。

支持向量机测试结果:

以 10000 个样本作为测试集,进行测试,得到数据如下:

样本数	1000	5000	10000	20000	30000	40000
正确率	88.73%	92.68%	93.41%	93.33%	93.45%	93.45%
时间	1.433s	2.083s	3.380s	9.820s	18.040s	29.070s

经过试验后发现,训练集样本容量增加超过 10000 后,识别准确率可达到 93%左右。随着样本集容量增加,训练时间会递增,每增加 10000 个样本,训练时间将增加超过 9 秒左右,而正确率没有明显的提升。支持向量机算法样本数在较少的时候就可以获得较高的正确率, 虽然相比于线性分类器运行时间较长,但相比于 BP 神经网络依然还是比较快捷的方法。

5 总结与展望

线性分类器具有实现简单, 适用范围广, 运算速度快等优点,作为模式识别最基础的分类器,它依然拥有较好的使用性能。本文选用 matlab 作为研究载体,研究了线性分类器,聚类分析,BP 神经网络,SVM 算法,比较它们的分类性能,得到以下结果:

5.1 在准确性上,SVM 算法最为优秀,BP 神经网络和线性分类器也可以达到较高正确率。而线性分类器的性能可以接受,且算法简单,容易实现,可以在简单案例中进行使用。

5.2 在运算速度时,BP 神经网络需要很长的训练时间,运算速度也不是很快, 所以在一些要求快速且准确的识别运算里,一般会选取 SVM 算法,所以在要求快速识别,较低要求的正确率时,我们可以选用线性识别器。

5.3 在要求精密识别的情况下,SVM 算法以较高的成功率会是第一选择。如果有需要多次使用, 有较长时间去训练且没有固定的概率模型时,我们可以使用 BP 神经网络。

模式识别在近几年得到了应用层面的快速发展, 本文进行的实验和比较只是模式识别领域的一部分算法的粗浅应用,在未来发展及投入实际应用时,还有更进步的算法和多组合器的方法可以改进整体性能和效率。