目录

kNN 与 Fisher 判别准则

1 kNN 算法

1.1 问题描述

问题背景：

有两组二维数据，在空间中的样本分布。对一个新的样本点，请尝试用 KNN 算法判断它的所属组别 。

提供数据：trainingData.mat

training：200x2 的矩阵，每行表示一个样本点

group：200x1 的矩阵，表示每个样本点的组别（1 或 2）

具体要求：

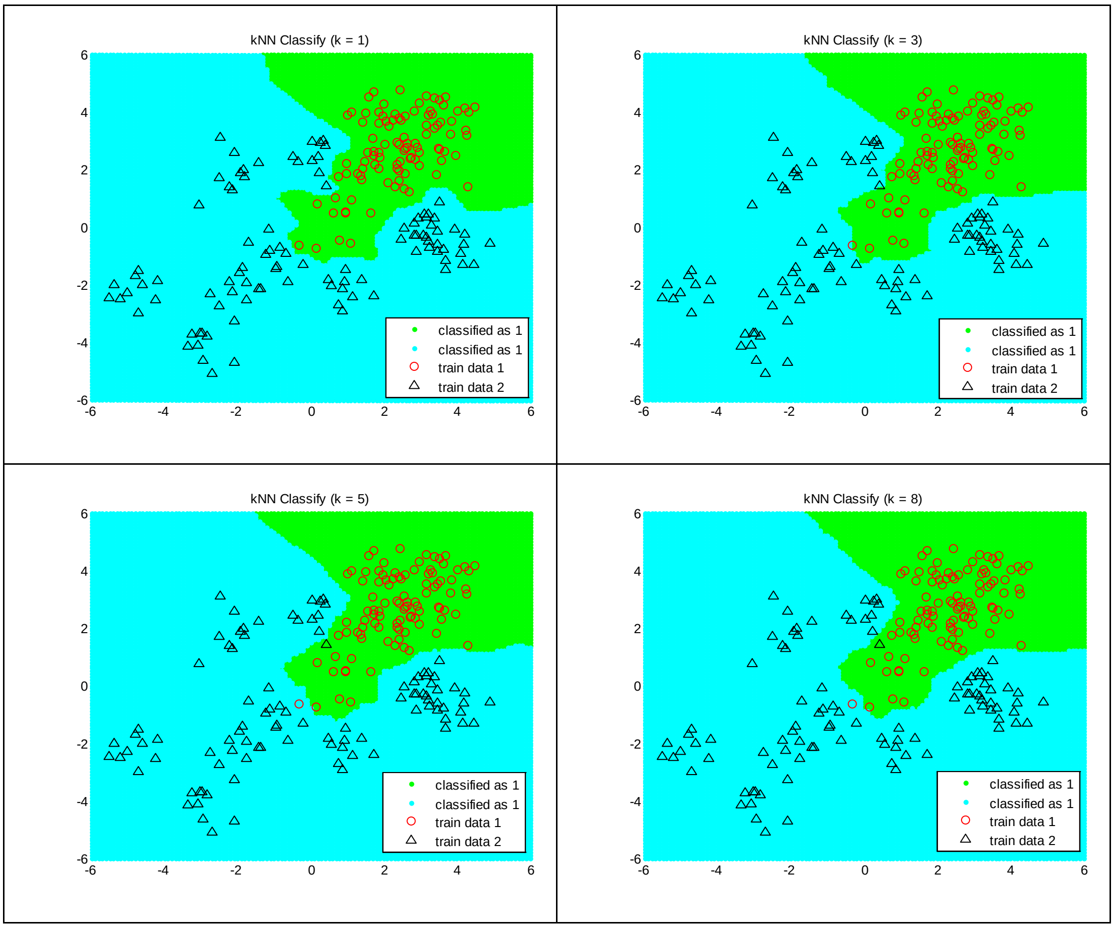
测试样本点集为{(x,y)|x=-6:0.1:6, y=-6:0.1:6}

尝试采用不同的 k 值（如 1、3、5），观察结果的变化并进行分析

1.2 仿真结果

kNN 算法的基本思路是：选择未知样本一定范围内确定个数的 k 个样本，该 k 个样本大多数属于某一类型，则未知样本判定为该类型。

仿真中，测试集是{(x,y)|x=-6:0.1:6, y=-6:0.1:6}，k 的值依次取[1 3 5 8 10 30 50 80]，得到的结果如表 1 所示。



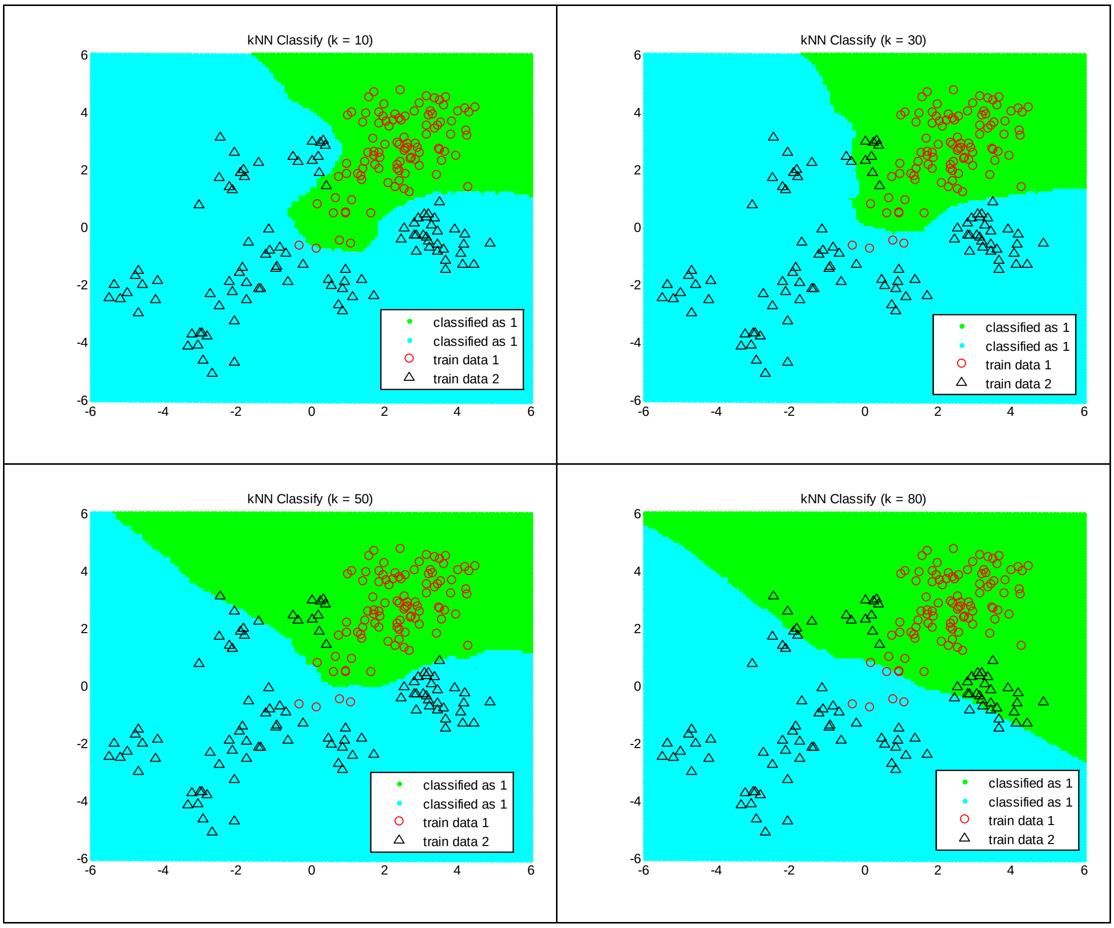


表 1 k 取不同值时的分类结果

1.3 结果分析

对 kNN 的分析如下：

1) 整体来看，k 值越大，分界面越平滑。k 很小时，分类容易过拟合；k 很大时，分类容易欠拟合。
2) k 值选取过小，由于得到的近邻数过少，会降低分类精度，同时也会放大噪声或数据的影响；而 k 选取过大时，如果训练集包含较少的类（本次实验就是这种情况）时，在选取 k 近邻时会把很多不相似的数据也包含进来，也会导致分类效果的降低。经验中，一般选取 k 略低于训练样本数的平方根。
3) kNN 算法属于实例学习，是 lazy learning 的一种，它在训练阶段不做任何事，而是在新实例到来时开始计算，分类。这导致该算法在分类时的开销非常大，因为要扫描全部样本计算距离。这点通过仿真也可以感觉到，如果不借助 Matlab 矩阵计算进行优化，计算过程往往耗时很久。
4) 可以考虑的对 kNN 的改进有
   a) 按照距离对 k 个最近邻进行加权，距离最近的权重越大。
   b) 不同粒度的 kNN，例如可以选取质心作为代表点，选出距离最近的一个或若干组，再在组内应用 kNN 算法，这样在大样本时能节省很多计算量。

2 Fisher 判别准则

2.1 问题描述

对给出的两种情况，求采用 Fisher 判别准则时的投影向量 W 和分类界面，并作图。

2.2 算法简述

在训练样本 X 的 n 维空间，获取与分类有关的参数：

1) 各类样本的均值向量

$$M_i = \frac{1}{N_i} \sum_{X \in \omega_i} X \tag{1}$$

2) 各类样本类内离散度矩阵，总类内离散度矩阵

$$S_i = \sum_{X \in \omega_i} \left(X - M_i\right)\left(X - M_i\right)^T \tag{2}$$

$$S_W = S_1 + S_2 \tag{3}$$

3) 投影向量

$$W^* = \frac{R}{\lambda} S_W^{-1} \left(M_1 - M_2\right) \tag{4}$$

4) 忽略比例因子，归一化处理

$$W_0 = \frac{W^*}{\left\|W^*\right\|} = \frac{S_W^{-1}\left(M_1 - M_2\right)}{\left\|S_W^{-1}\left(M_1 - M_2\right)\right\|} \tag{5}$$

实际上，Fisher 准则下 X 的 n 维空间向 Y 的一维空间投影的最佳方向，是向量 $\left(M_1 - M_2\right)$ 旋转 S_W^{-1}，即根据总类内离散度矩阵修正两类中心连线方向。

2.3 仿真结果

两种情况对应的分类结果如图 1 和图 2 所示。

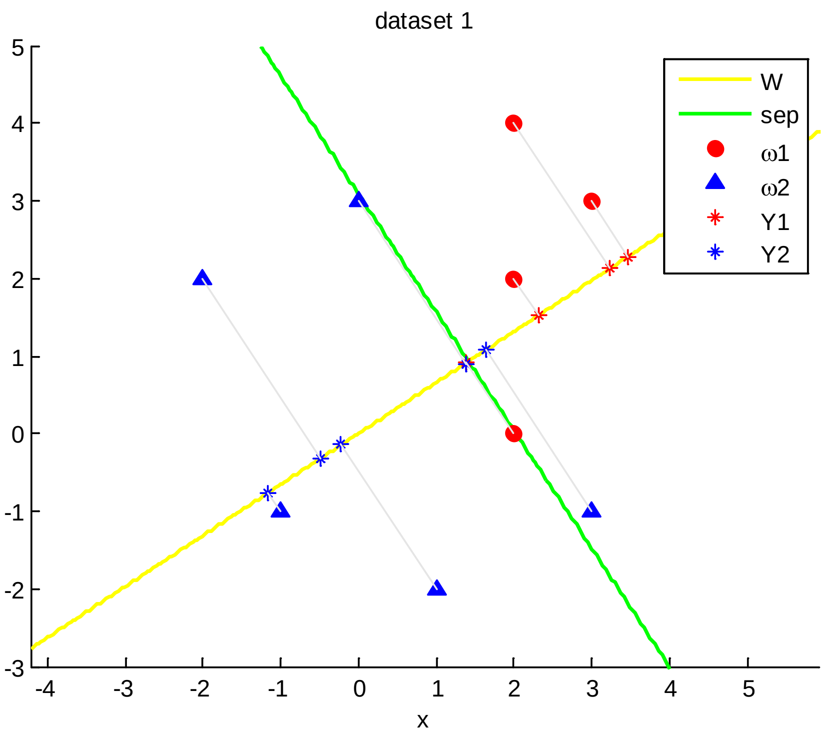


图 1 数据集 1 的分类结果

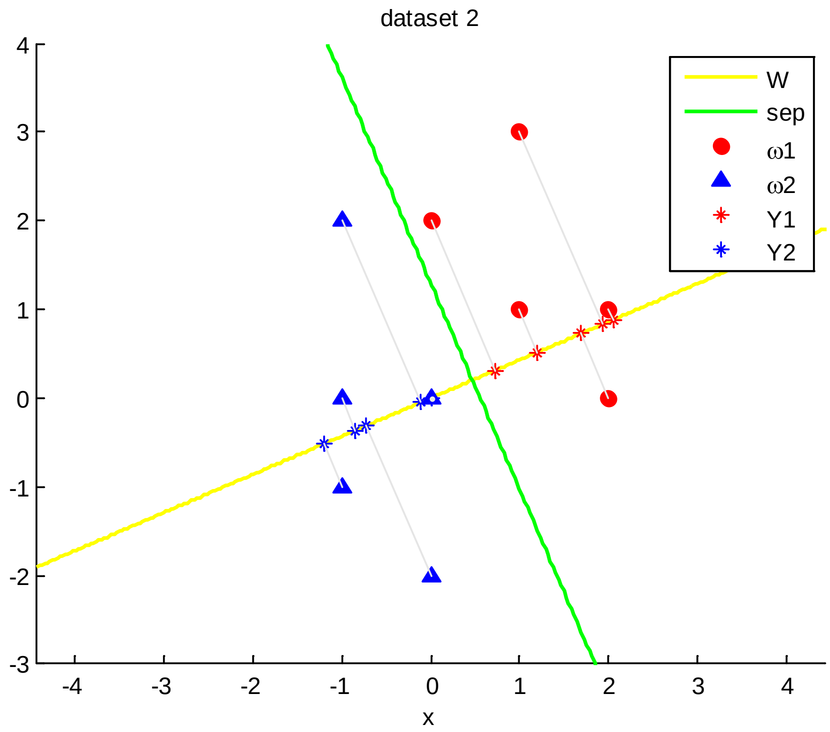


图 2 数据集 2 的分类结果

2.4 结果分析

之前提到的无论是最小距离判别，还是线性识别函数方法，本质上都是向一个一维向量投影，将 n 维空间的全体样本投影到一条直线上进行分类，而 Fisher 准则实际上是在找这样的一条直线，它使得投影后分开得最好。

因此，从线性意义上来讲，只要数据是线性可分的，Fisher 准则一定能够找到一个能正确分类的分界面，而且这个分界面是线性投影意义上最优的。

仿真样本中，数据集 1 不是线性可分的，因而没有能找到一个分界面完全正确分类，但数据集 2 是线性可分的，从图 2 也可以看出，两类数据被很好地分开了。

上次实验中提到，由于迭代求取权向量算法是基于迭代的思想，一旦程序找到了能正确分类的分界面即终止迭代，这样往往导致分界面与其中某一类数据集边界非常接近。由于 Fisher 准则着眼于寻找最优的投影直线，因此可以很好地避免这个问题。