机器学习课程实验四

2022 年 10 月 20 日　苏博南　202000460020

1　二分类的 Linear Discrimination Analysis

相关原理省略，只考虑算法流程。记 x 为一 $m \times n$ 的矩阵表示训练集合，C_i 表示第 i 个类别,$i = 1, 2$。则:

1. 计算平均点：$\mu_i = \frac{1}{n_i}\sum_{x\in C_i} x, i = 1, 2$，其中 n_i 表示类别为 C_i 的样本数量。
2. 计算类内散度矩阵：$S_w = \sum_{i=1}^{C}\sum_{x\in C_i}(x - \mu_i)(x - \mu_i)^T$.
3. 计算投影向量：$\theta^* = S_w^{-1}(\mu_1 - \mu_2)$
4. 作出每个样本点在直线 $\overrightarrow{\theta^*}$ 上的投影。

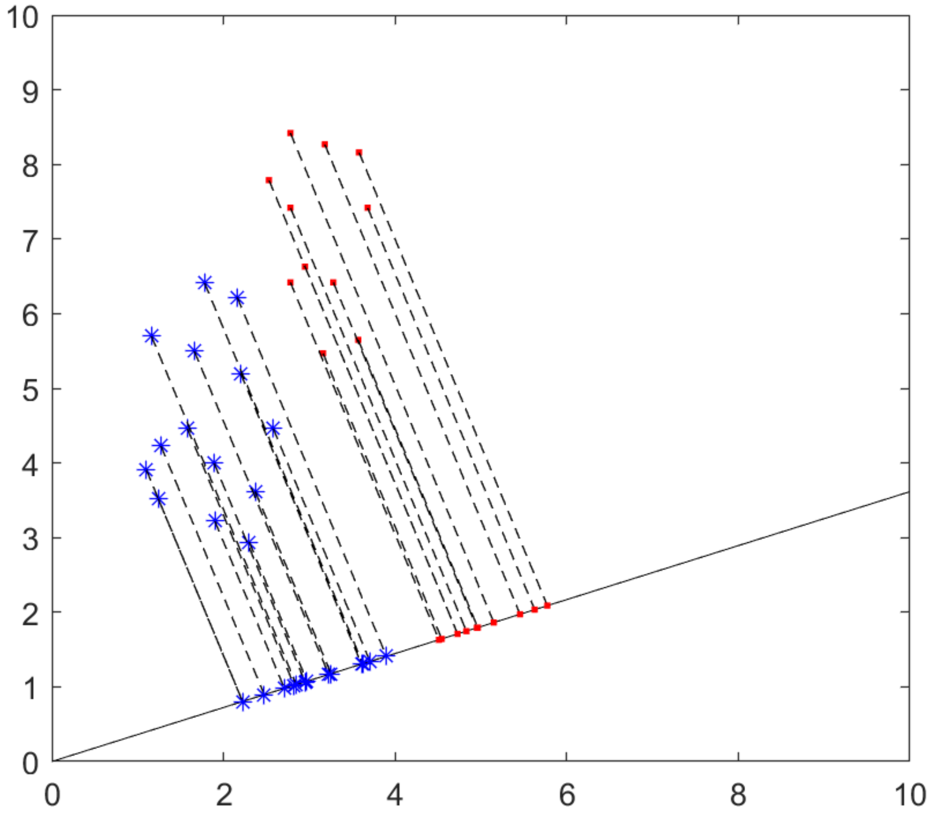
在给出一个预测样本 x，只需要计算 $y = \theta^{*T}x$ 然后将 y 与 $\gamma = \frac{n_1\theta^{*T}\mu_1 + n_2\theta^{*T}\mu_2}{n_1+n_2}$ 比较，得到分类结果。

算法代码如下:

```
blue = load("ex3Data/ex3blue.dat");
red = load("ex3Data/ex3red.dat");

plot(blue(:, 1), blue(:, 2), 'b*');
hold on;
plot(red(:, 1), red(:, 2), 'r.');
axis([0, 10, 0, 10]);

mu_blue = mean(blue)';
mu_red = mean(red)';

[m, n] = size(blue);
S_w = zeros(n, n);
for i = 1 : m
    S_w = S_w + (blue(i, :)' - mu_blue) * (blue(i, :)' - mu_blue)';
end
[m, n] = size(red);
for i = 1 : m
    S_w = S_w + (red(i, :)' - mu_red) * (red(i, :)' - mu_red)';
end

theta = S_w \ (mu_blue - mu_red);

xs = 0 : 0.1 : 10;
ys = theta(2) / theta(1) * xs;
plot(xs, ys, 'k-');

for i = 1 : size(blue, 1)
    p = projects(theta, blue(i, :)');
```

```
  line([blue(i, 1), p(1)], [blue(i, 2), p(2)], 'LineStyle', '--', 'Color', 'k', 'LineWidth', 0.5);
  plot(p(1), p(2), 'b*');
end

for i = 1 : size(red, 1)
  p = projects(theta, red(i, :)');
  line([red(i, 1), p(1)], [red(i, 2), p(2)], 'LineStyle', '--', 'Color', 'k', 'LineWidth', 0.5);
  plot(p(1), p(2), 'r.');
end

function pos = projects(line, point)
  t = (point' * line) / (line' * line);
  pos = t * line;
end
```

得到的结果如下图所示:



2 多分类的 LDA

算法流程:

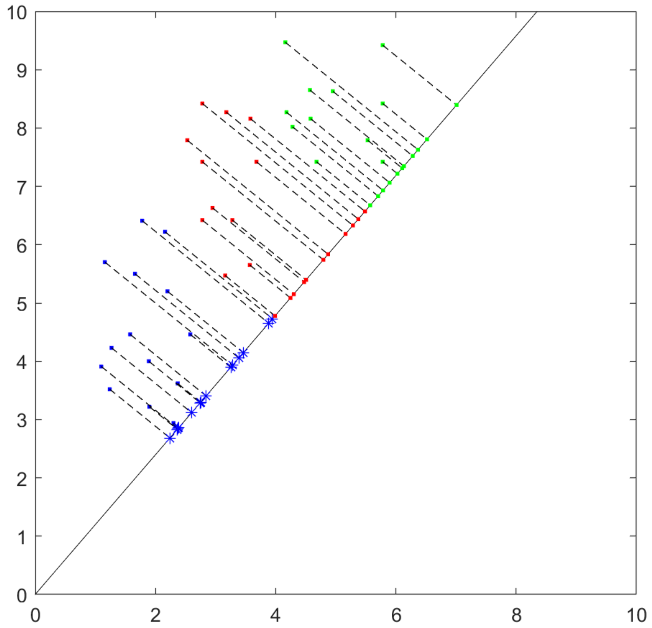
1. 计算平均点 $\mu_i = \frac{1}{n_i}\sum_{x\in C_i} x$。
2. 计算类间散度矩阵 $S_b = \frac{1}{2N}\sum_{i,j=1}^{C} n_i n_j(\mu_i - \mu_j)(\mu_i - \mu_j)^T$
3. 计算类内散度矩阵 $S_w = \sum_{i=1}^{C}\sum_{x\in C_i}(x-\mu_i)(x-\mu_i)^T$
4. 选择 $S_w^{-1}S_b$ 的前 p 大的特征值对应的特征向量 $\{\theta_1, \theta_2, ..., \theta_p\}$。

在实验中，由于样本数据维度为两维，只有两个特征值，我们选择其中大的哪一个。算法代码如下：

```
blue = load("ex3Data/ex3blue.dat");
red = load("ex3Data/ex3red.dat");
green = load("ex3Data/ex3green.dat");

plot(blue(:, 1), blue(:, 2), 'b.');
hold on;
plot(red(:, 1), red(:, 2), 'r.');
plot(green(:, 1), green(:, 2), 'g.');

axis([0, 10, 0, 10]);

mu_blue = mean(blue)';
mu_red = mean(red)';
mu_green = mean(green)';

mu = (sum(blue) + sum(red) + sum(green)) / (size(blue, 1) + size(red, 1) + size(
green, 1));
mu = mu';

S_b = size(blue, 1) * (mu_blue - mu) * (mu_blue - mu)' + size(red, 1) * (mu_red
- mu) * (mu_red - mu)' + size(green, 1) * (mu_green - mu) * (mu_green - mu)';

S_w = zeros(2, 2);
for i = 1 : size(blue, 1)
    S_w = S_w + (blue(i, :) - mu_blue) * (blue(i, :) - mu_blue)';
end
for i = 1 : size(red, 1)
    S_w = S_w + (red(i, :) - mu_red) * (red(i, :) - mu_red)';
end
for i = 1 : size(green, 1)
    S_w = S_w + (green(i, :) - mu_green) * (green(i, :) - mu_green)';
end

S = S_w \ S_b;
[V, D] = eig(S);
[mx, i] = max(diag(D));
theta = V(:, i);

xs = 0 : 0.1 : 10;
ys = theta(2) / theta(1) * xs;
```

```
    plot(xs, ys, ’k−’);

    for i = 1 : size(blue, 1)
        p = projects(theta, blue(i, :)');
        line([blue(i, 1), p(1)], [blue(i, 2), p(2)], ’LineStyle’, ’−−’, ’Color’, ’k’, ’LineWidth’, 0.5);
        plot(p(1), p(2), ’b∗’);
    end

    for i = 1 : size(red, 1)
        p = projects(theta, red(i, :)');
        line([red(i, 1), p(1)], [red(i, 2), p(2)], ’LineStyle’, ’−−’, ’Color’, ’k’, ’LineWidth’, 0.5);
        plot(p(1), p(2), ’r.’);
    end

    for i = 1 : size(green, 1)
        p = projects(theta, green(i, :)');
        line([green(i, 1), p(1)], [green(i, 2), p(2)], ’LineStyle’, ’−−’, ’Color’, ’k’, ’LineWidth’, 0.5);
        plot(p(1), p(2), ’g.’);
    end

    function pos = projects(line, point)
        t = (point' * line) / (line' * line);
        pos = t * line;
    endblue = load("ex3Data/ex3blue.dat");
    red = load("ex3Data/ex3red.dat");
    green = load("ex3Data/ex3green.dat");

    plot(blue(:, 1), blue(:, 2), ’b.’);
    hold on;
    plot(red(:, 1), red(:, 2), ’r.’);
    plot(green(:, 1), green(:, 2), ’g.’);

    axis([0, 10, 0, 10]);

    mu_blue = mean(blue)';
    mu_red = mean(red)';
    mu_green = mean(green)';

    mu = (sum(blue) + sum(red) + sum(green)) / (size(blue, 1) + size(red, 1) + size(green, 1));
    mu = mu';

    S_b = size(blue, 1) * (mu_blue - mu) * (mu_blue - mu)' + size(red, 1) * (mu_red - mu) * (mu_red - mu)' + size(green, 1) * (mu_green - mu) * (mu_green - mu)';

    S_w = zeros(2, 2);
    for i = 1 : size(blue, 1)
        S_w = S_w + (blue(i, :) - mu_blue) * (blue(i, :) - mu_blue)';
    end
    for i = 1 : size(red, 1)
        S_w = S_w + (red(i, :) - mu_red) * (red(i, :) - mu_red)';
    end
    for i = 1 : size(green, 1)
        S_w = S_w + (green(i, :) - mu_green) * (green(i, :) - mu_green)';
    end

    S = S_w \ S_b;
    [V, D] = eig(S);
    [mx, i] = max(diag(D));
    theta = V(:, i);

    xs = 0 : 0.1 : 10;
    ys = theta(2) / theta(1) * xs;
    plot(xs, ys, 'k-');

    for i = 1 : size(blue, 1)
        p = projects(theta, blue(i, :)');
        line([blue(i, 1), p(1)], [blue(i, 2), p(2)], 'LineStyle', '--', 'Color', 'k', 'LineWidth', 0.5);
        plot(p(1), p(2), 'b*');
    end

    for i = 1 : size(red, 1)
        p = projects(theta, red(i, :)');
        line([red(i, 1), p(1)], [red(i, 2), p(2)], 'LineStyle', '--', 'Color', 'k', 'LineWidth', 0.5);
        plot(p(1), p(2), 'r.');
    end

    for i = 1 : size(green, 1)
        p = projects(theta, green(i, :)');
        line([green(i, 1), p(1)], [green(i, 2), p(2)], 'LineStyle', '--', 'Color', 'k', 'LineWidth', 0.5);
        plot(p(1), p(2), 'g.');
    end

    function pos = projects(line, point)
        t = (point' * line) / (line' * line);
        pos = t * line;
    end
```

得到的结果如下图所示:


0
1
2
3
4
5
6
7
8
9
10
0
2
4
6
8
10