homework 2

Zuyao Chen 201728008629002

1 Problem 1

对于多类情况 1,M 类需要 M 个判别函数; 多类情况 2,M 类需要 $M(M-1)/2$ 个判别函数, 因此所需判别函数的最少数目为

$$3 + 7 * (7-1)/2 = 24$$

2 Problem 2

$$d_1(x) = -x_1, d_2(x) = x_1 + x_2 - 1, d_3(x) = x_1 - x_2 - 1$$

1. 多类模式 1, 下图中白色区域表示不确定性区域

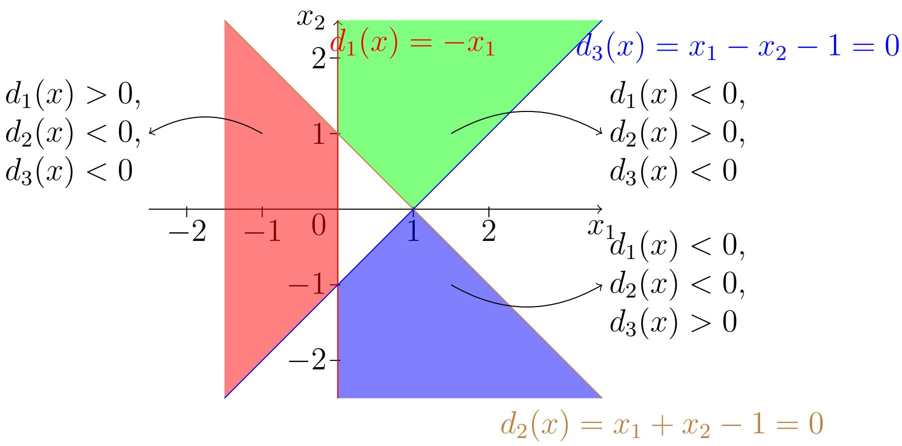

Figure 1: 多类情况 1 的判别界面示意图

2. 多类模式 2, 下图中白色区域表示不确定性区域

$$d_{12}(x) = d_1(x), d_{13}(x) = d_2(x), d_{23}(x) = d_3(x)$$

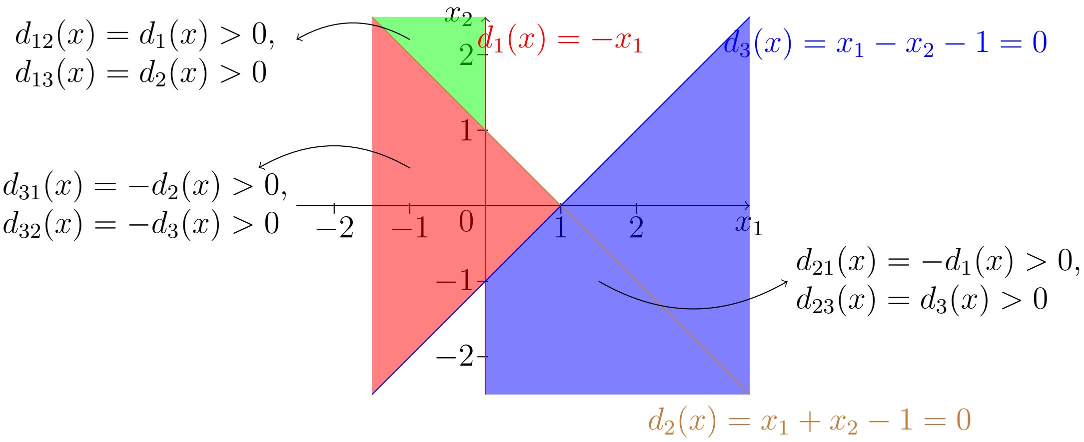

Figure 2: 多类情况 2 的判别界面示意图

3. 多类情况 3 的条件下,

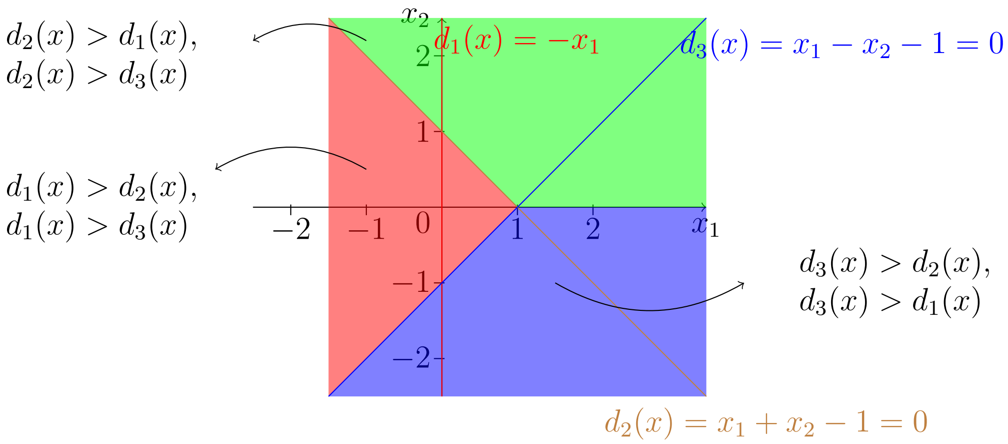

Figure 3: 多类情况 3 的判别界面示意图

3 Problem 3

若数据线性可分, 则至少需要 $C_4^1 = 4$ 个系数分量; 若要建立二次的多项式判别函数, 则至少需要 $C_5^2 = 10$ 个系数分量.

4 Problem 4

Q: 用感知器算法求下列模式分类的解向量 w:

$$w_1 : (0,0,0)^T, (1,0,0)^T, (1,0,1)^T, (1,1,0)^T$$
$$w_2 : (0,0,1)^T, (0,1,1)^T, (0,1,0)^T, (1,1,1)^T$$

A: 二分类的感知机算法训练过程如下:

- 线性分类器设为 $h_{\boldsymbol{w}}(\boldsymbol{x}) = \text{sign}(\boldsymbol{w}^T\boldsymbol{x} + b)$, sign 为符号函数, 如果 $\boldsymbol{w}^T\boldsymbol{x} + b \geq 0$ 那么 $h = 1$ 否则 $h = -1$;

- 分类损失 loss 定义为分类错误加 1, 分类正确不变;
- 扩充每个样本 $\boldsymbol{x}$ 为 $[\boldsymbol{x}, 1]$ 的向量, 参数设为 $\boldsymbol{\theta} = [\boldsymbol{w}, b]$, 参数更新规则:

$$\boldsymbol{\theta}_j = \boldsymbol{\theta}_j + \alpha(y^{(i)} - h_{\boldsymbol{\theta}}(x^{(i)}))\boldsymbol{x}_j^{(i)}, i = 1, 2, ..., m, j = 1, 2, ..., n$$

  其中 m 为样本数量, n 为参数的维度, α 为学习速率.
- 初始化参数进行迭代, 当总的分类损失为 0 就停止训练得到一个线性分类器, 若 loss 不能降为 0 则表明有的样本未分类正确, 数据不是线性可分.

程序及测试结果如下

```python
#!/usr/bin/env python

import numpy as np
'''
perceptron algorithm
1.hypothesis function: h_w(x) = sign(w'x+b)
sign(x) = 1, if x >= 0 otherwise sign(x) = -1
2.update parameters:
 w_j = w_j + lr*( y^i- h_w(x^i))*x_j^i

'''
max_iters = 100000
lr = 1
def hypothesis(x, theta):
  if np.dot(theta, x) >= 0:
    return 1
  else:
    return -1

def total_loss(x, y, theta):
  out = 0
  for i in range(len(x)):
    h = hypothesis(x[i], theta)
    if not h == y[i]:
      out += 1
  return out

def train(X,Y,theta):
  loss = total_loss(X,Y,theta)
  print 'learning rate:', lr
  print 'initial theta:', theta
  print 'initial loss:', loss
  for it in range(max_iters):
    for i in range(len(X)): #a training set
      h = hypothesis(X[i], theta)
      theta += lr*(Y[i] - h)*X[i]

    loss = total_loss(X,Y, theta)
    if __name__ == '__main__':
      print 'iter:', it
      print 'theta:', theta
      print 'loss:', loss
    if loss == 0:
      print 'optimization finished!'
      return
```

```
  if loss != 0:
    print 'not linearly separable!'
  return

if __name__ == '__main__':
  w1 = [[0,0,0],[1,0,0],[1,0,1],[1,1,0]] #label y = 1
  w2 = [[0,0,1],[0,1,1],[0,1,0],[1,1,1]] #label y = -1
  y1 = [1]*len(w1)
  y2 = [-1]*len(w2)
  X = np.concatenate((w1,w2),axis = 0)
  b = np.ones(len(X))
  b.shape = (len(b), 1)
  X = np.concatenate((X,b), axis = 1)
  Y = np.concatenate((y1,y2),axis = 0)
  W = np.random.normal(0,0.01,4)


  train(X,Y,W)
  print 'theta:', W
```

测试结果:

```
learning rate: 1
initial theta: [-0.00948958 -0.00704727 -0.01104384  0.00286872]
initial loss: 3
iter: 0
theta: [ 1.99051042 -0.00704727 -2.01104384  0.00286872]
loss: 1
iter: 1
theta: [ 1.99051042 -2.00704727 -4.01104384 -1.99713128]
loss: 4
iter: 2
theta: [  1.99051042e+00  -4.00704727e+00  -4.01104384e+00   2.86871966e-03]
loss: 2
iter: 3
theta: [ 3.99051042 -4.00704727 -2.01104384  2.00286872]
loss: 0
optimization finished!
theta: [ 3.99051042 -4.00704727 -2.01104384  2.00286872]
```

3 次迭代得到一个参数 $\theta = [3.99051042, -4.00704727, -2.01104384, 2.00286872]$, 判别函数

$$D(x) = 3.99051042x_1 - 4.00704727x_2 - 2.01104384x_3 + 2.00286872$$

5 Problem 5

Q: 用多类感知器算法求下列模式的判别函数:

$$\begin{aligned} w_1 &:(-1,-1)^T \\ w_2 &:(0,0)^T \\ w_3 &:(1,1)^T \end{aligned}$$

A:
依次选取一类样本作为正样本, 其余作为负样本, 进行二分类的感知机训练, 得到 3 个判别函数. 程序及测试结果如下

```
#!/usr/bin/env python
```

```python
import numpy as np
from perceptron import train

'''
for multi class, choose one as positive sample, others are negative samples
'''

if __name__ == '__main__':
  D = 2
  w1 = [[-1,-1]] #
  w2 = [[0,0]]   #
  w3 = [[1,1]]
  y2 = [-1]*len(w2)
  y3 = [-1]*len(w3)
  #choose w1 as positive samples
  y1 = [1] * len(w1)
  X = np.concatenate((w1,w2,w3),axis = 0)
  b = np.ones(len(X))
  b.shape = (len(b), 1)
  X = np.concatenate((X,b), axis = 1)
  Y = np.concatenate((y1,y2,y3),axis = 0)

  theta1 = np.random.normal(0,0.01, 1+D) # parameters
  train(X,Y,theta1)
  print 'theta1:', theta1
  #choose w2 as positive samples
  y1 = [-1]*len(w1)
  y2 = [1]*len(w2)
  Y = np.concatenate((y1,y2,y3),axis = 0)
  theta2 = np.random.normal(0,0.01, 1+D)
  train(X,Y,theta2)
  print 'theta2:', theta2
  #choose w3 as positive samples
  y2 = [-1]*len(w2)
  y3 = [1]*len(w3)
  Y = np.concatenate((y1,y2,y3), axis = 0)
  theta3 = np.random.normal(0,0.01, 1+D)
  train(X,Y,theta3)
  print 'theta3:', theta3
```

测试结果:

```
learning rate: 1
initial theta: [-0.00887894  0.01173659  0.00866764]
initial loss: 2
optimization finished!
theta1: [-2.00887894 -1.98826341 -1.99133236]
learning rate: 1
initial theta: [-0.01577603 -0.00699137  0.00682586]
initial loss: 1
not linearly separable!
theta2: [-0.01577603 -0.00699137  0.00682586]
learning rate: 1
initial theta: [ 0.00662045 -0.00524078  0.00407962]
initial loss: 2
optimization finished!
theta3: [ 2.00662045  1.99475922 -1.99592038]
```

对于类别 w_1, 线性可分, 一个判别函数为

$$d_1(x) = -2.00887894x_1 - 1.98826341x_2 - 1.99133236$$

对于类别 w_2, 线性不可分, 判别函数

$$d_2(x) = 0$$

对于类别 w_3, 线性可分, 一个判别函数

$$d_3(x) = 2.00662045x_1 + 1.99475922x_2 - 1.99592038$$

6 Problem 4

Q: 采用梯度法和准则函数

$$J(w, x, b) = \frac{1}{8||x||^2}[(w^T x - b) - |w^T x - b|]^2$$

式中实数 b>0，试导出两类模式的分类算法。
A:

$$J(w, x, b) = \begin{cases} 0, \text{ if } w^T x - b > 0 \\ \dfrac{1}{2||x||^2}(w^T x - b)^2, \text{ otherwise} \end{cases}$$

J 对 w 求导得

$$\frac{\partial J}{\partial w} = \begin{cases} 0, \text{ if } w^T x - b > 0 \\ \dfrac{1}{||x||^2}(w^T x - b)x, \text{ otherwise} \end{cases}$$

对 b 求导, 得

$$\frac{\partial J}{\partial b} = \begin{cases} 0, \text{ if } w^T x - b > 0 \\ -\dfrac{1}{||x||^2}(w^T x - b), \text{ otherwise} \end{cases}$$

参数更新规则

- 初始化 w,b
- 选取一个样本 x, 计算 $h = w^T x - b$, 若 $h > 0$ 则参数不变, 继续选取一个样本; 否则, 计算梯度, 更新参数:

$$\begin{aligned} w :=& w + \alpha \frac{1}{||x||^2}(b - w^T x)x, \\ b :=& b + \alpha \frac{1}{||x||^2}(w^T x - b), \alpha > 0 \end{aligned}$$

- 如果模式线性可分, 则当所有样本均满足 $w^T x - b > 0$ 时停止训练; 否则得不到收敛结果.

7 Problem 5

Q: 用二次埃尔米特多项式的势函数算法求解以下模式的分类问题

$$w_1 : (0, 1)^T, (0, -1)^T$$
$$w_2 : (1, 0)^T, (-1, 0)^T$$

A:
选择 Hermite 多项式, 其正交域为 $(-\infty, +\infty)$, 其一维形式为

$$\varphi_k = \frac{\exp(-x^2/2)}{\sqrt{2^k \cdot k!\sqrt{\pi}}} H_k(x), k = 0, 1, 2...$$
$$H_n(x) = (-1)^n \exp(x^2) \frac{\mathrm{d}^n}{\mathrm{d}x^n} \exp(-x^2)$$

其正交性

$$\int_{-\infty}^{\infty} H_m(x) H_n(x) \exp(-x^2) \mathrm{d}x = \begin{cases} 0, m \neq n \\ 2^n n! \sqrt{\pi}, m = n \end{cases}$$

其中，$H_k(x)$ 前面的乘式为正交归一化因子，为计算简便可省略。因此，Hermite 多项式前面几项的表达式为

$$H_0(x) = 1, H_1(x) = 2x, H_2(x) = 4x^2 - 2,$$
$$H_3(x) = 8x^3 - 12x, H_4(x) = 16x^4 - 48x^2 + 12$$

建立二维的正交函数集, 这里取 9 项最低阶的二维的正交函数

$$\begin{aligned}
\varphi_1(x) &= \varphi_1(x_1, x_2) = H_0(x_1)H_0(x_2) = 1 \\
\varphi_2(x) &= \varphi_2(x_1, x_2) = H_1(x_1)H_0(x_2) = 2x_1 \\
\varphi_3(x) &= \varphi_3(x_1, x_2) = H_0(x_1)H_1(x_2) = 2x_2 \\
\varphi_4(x) &= \varphi_4(x_1, x_2) = H_1(x_1)H_1(x_2) = 4x_1x_2 \\
\varphi_5(x) &= \varphi_5(x_1, x_2) = H_0(x_1)H_2(x_2) = 4x_2^2 - 2 \\
\varphi_6(x) &= \varphi_6(x_1, x_2) = H_1(x_1)H_2(x_2) = 2x_1(4x_2^2 - 2) \\
\varphi_7(x) &= \varphi_7(x_1, x_2) = H_2(x_1)H_0(x_2) = 4x_1^2 - 2 \\
\varphi_8(x) &= \varphi_8(x_1, x_2) = H_2(x_1)H_1(x_2) = (4x_2^2 - 2)2x_2 \\
\varphi_9(x) &= \varphi_9(x_1, x_2) = H_2(x_1)H_2(x_2) = (4x_1^2 - 2)(4x_2^2 - 2)
\end{aligned}$$

按第一类势函数的定义生成势函数,

$$K(x, x_k) = \sum_{i=1}^{9} \varphi_i(x)\varphi_i(x_k)$$

$$d(x) = -32x_1^2 + 32x_2^2$$

8 Problem 6

$$\begin{aligned}
K(x, x_k) &= e^{-\alpha[(x_1 - x_{k1})^2 + (x_2 - x_{k2})^2]} \\
d(x) &= e^{-\alpha}[e^{-x_1^2 - (x_2+1)^2} + e^{-x_1^2 - (x_2-1)^2} - e^{-(x_1-1)^2 - x_2^2} - e^{-(x_1+1)^2 - x_2^2}]
\end{aligned}$$