

```

# 13
using Statistics
using Random
x = []
x0=1.0
datafile = open("train.txt","r") #read the data
lines = readlines(datafile)
close(datafile)
for (i, line) in enumerate(lines)
    tmp = parse.(Float64, split(line,"\t"))
    insert!(tmp,1,x0)
    append!(x,tmp)
end

m = reshape(x,(12,100))
m = m'
X = m[:,1:11]
X = convert(Matrix{Float64}, X)
y = m[:,12]

w_lin = inv(X'*X)*X'*y

#13
println(mean((X*w_lin .- y) .^2))
# 0.792234776110557

```

```

# 14, 15, 16, 17, 18
function logistic(x)
    return 1 / (1 + exp(-x))
end

x_test=[]
datafile = open("test.txt","r") #read the data
lines = readlines(datafile)
close(datafile)
for (i, line) in enumerate(lines)
    tmp = parse.(Float64, split(line,"\t"))
    insert!(tmp,1,x0)
    append!(x_test,tmp)
end

m_test = reshape(x_test,(12,400))
m_test = m_test'
X_test = m_test[:,1:11]
X_test = convert(Matrix{Float64}, X_test)
y_test = m_test[:,12]

function SGD(;mode = 1, w0 = 1, test=0)

    eta = 0.001
    err = [0.0 for i in 1:1000]
    diff_err = 0
    w = [0 for i in 1:11]
    for _i in 1:1000

        Random.seed!(_i)
        if w0 == 1
            w = [0 for i in 1:11]
        else
            w = w_lin
        end

        for _j in 1:800

            ranum = Int(ceil(rand()*1000%100)) #100 train smaple
            ver_x = X[ranum,:]

            if mode==1 || mode==3
                w = w .+ y[ranum]*X[ranum,:] *eta *logistic(-y[ranum]*w'*X[ranum, :])
            elseif mode==2
                w = w .- eta * 2 * (ver_x*ver_x'*w - ver_x*y[ranum])
            end
        end

        if mode==1

```

```

err[_i] = mean(log.(exp.(-y.*X*w).+1))

elseif mode==2
    err[_i] = mean((X*w .- y).^2)

else mode==3 || mode==4
    cnt = count(sign.(X * w) .!= y )
    err[_i] = cnt/length(y)
end

if test==1
    cnt_test = count(sign.( X_test * w) .!= y_test)
    err[_i] = abs(err[_i]-cnt_test/length(y_test))
end

end

println("err: ",mean(err))
end

# parameter
# _mode_
# 1: SDG_Log + ce /2:SDG_linear + sqr_err /3: bi_err + SDG_Log /4:no_SGD + bi_err
# _w0_
# 1: zero /2: w = w_lin
# _test_
# 1: true /0: false

#14
SGD(mode = 2, w0 = 1, test = 0)
# 0.823023080631579

#15
SGD(mode = 1, w0 = 1, test = 0)
# 0.6571772493920419

#16
SGD(mode = 1, w0 = 2, test = 0)
# 0.6052071726706776

#17
SGD(mode = 3, w0 = 2, test = 1)
# 0.029912500000000012

#18
SGD(mode = 4, w0 = 2, test = 1)
# 0.040000000000000003

```

```

# 19, 20
function p_transform(;order = 1)
    X_p = m[:,2:11]
    X_final = [1 for i in 1:length(y)]

    X_test_p = m_test[:,2:11]
    X_test_final = [1 for i in 1:length(y_test)]
    for i in 1:order
        tmp = X_p.^i
        X_final = hcat(X_final, tmp)

        tmp = X_test_p.^i
        X_test_final = hcat(X_test_final, tmp)
    end
    #@show size(X_final)
    w_lin = inv(X_final'*X_final)*X_final'*y

    cnt=0
    for i in 1:length(y)
        #@show sign(y[i]*w'*X[i,:])
        if sign(y[i]*w_lin'*X_final[i,:])!=1
            cnt = cnt + 1
        end
    end
    err = cnt/length(y)
    #@show err
    cnt_test = 0
    for i in 1:length(y_test)
        #@show y_test[i]*w'*X[i,:]
        if sign(y_test[i]*w_lin'*X_test_final[i,:])!=1
            cnt_test = cnt_test + 1
        end
    end
    #@show cnt_test/length(y_test)
    #@show abs(cnt_test/length(y_test))
    err = abs(err - cnt_test/length(y_test))
    println(err)
end

```

```

#19
p_transform(order = 2)
# 0.08249999999999999

```

```

#20
p_transform(order = 8)
# 0.415

```

