204789: Machine Learning and Neural Network


julia

Defining a function

- สามารถนิยามโดย

```
function name(arg1, arg2, ... )
    # function body
    return ret1, ret2, ...          # สามารถ return ได้มากกว่า 1 ค่า
end
```

- Function call

```
[a, b, c ... ]  =  name(arg1, arg2, ...)
```

Exercise 1

- ให้ลองสร้างฟังก์ชัน logisticRegresion()
- ฟังก์ชันนี้จะรับ input คือ
  - X = input data
  - Y = input label
  - W = weight vector
- ฟังก์ชันนี้จะคืนค่า otpimised W กลับมา
- ทดลอง classify boston.mat dataset เพื่อตรวจสอบความถูกต้อง

Exercise 2

- ใช้ Logistic Regression ที่สร้างไปข้างต้นผนวกเข้ากับ GaussianKernel
- สิ่งที่ได้จะคือ kernel logistic regression ที่สามารถ classify non-linear data ได้
- ทดลอง classify banana.mat data ว่า kernel LR ที่สร้างขึ้นมาได้ผลหรือไม่

Using the kernel

```
srand(123)
x = rand(5,2)
```

```
5x2 Array{Float64,2}:
 0.768448  0.662555
 0.940515  0.586022
 0.673959  0.0521332
 0.395453  0.26864
 0.313244  0.108871
```

X_1

```
k1 = kernelmatrix(K,x)
```

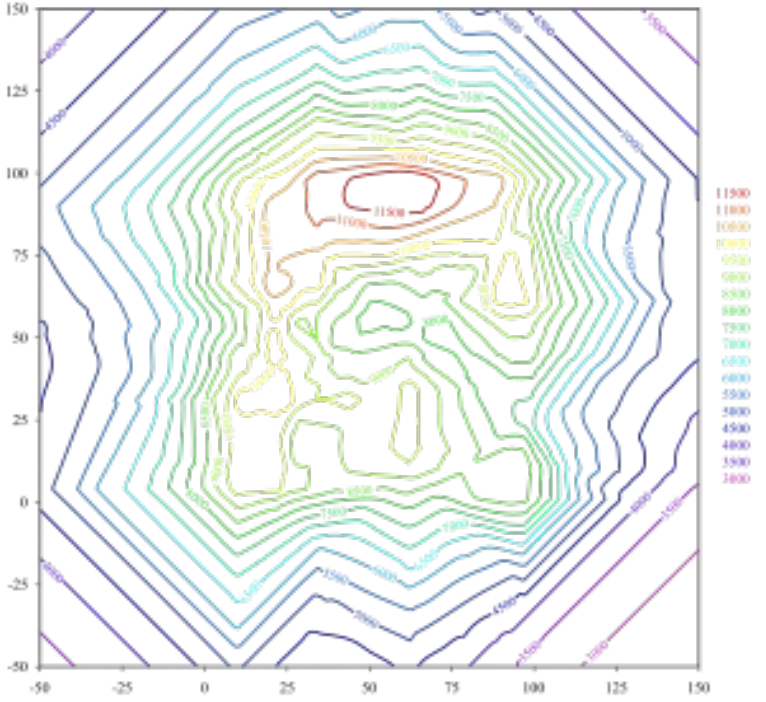
```
5x5 Array{Float64,2}:
 1.0       0.931528  0.466226  0.55511   0.357882
 0.931528  1.0       0.490574  0.451289  0.288724
 0.466226  0.490574  1.0       0.779673  0.765927
 0.55511   0.451289  0.779673  1.0       0.937472
 0.357882  0.288724  0.765927  0.937472  1.0
```

Contour package

- Contour is a curve along which function has a constant value (aka isoline)
- มีประโยชน์ในการ plot decision boundary
  - E.g., function value along the boundary is zero.
- Install the package

  Pkg.add(“Contour”)

  using Contour



Contour Example

- Want to plot a contour of f(x,y) = x^2+y^2

```
x = -3:0.01:3

y = -4:0.02:5

z = [Float64((xi^2 + yi^2)) for xi in x, yi in y]
```

- Now find the contours

```
c = contours(x,y,z)
```

Contour Example

- We can extract levels from contour ‘c’ using

  lv = levels(c) <= this returns an iterator

- Extract z value in that level using

  zlv = level(lv) <= this return float

- We can extract line segments in each level using

  ln = lines(lv) <= this also return an iterator

- We can extract end points of each line using

  xs, ys = coordinates(ln)

Contour: the whole thing

```
using Winston, Contour

# define x,y,z as above

# define FramedPlot()

for lv in levels(contours(x,y,z))

    zlv = level(lv)

    for line in lines(lv)

        xs, ys = coordinates(line)
add(p,Curve(xs,ys,color=Winston.default_color(Integer(floor(zlv)))))

    end

end

# call p to show the figure
```

Exercise 3

Plot the decision boundary of

1. Logistic regression on banana.mat
2. Kernel logistic regression on the same dataset

AdaBoost

- There is a package call ‘DecisionTree’ that implements adaboost using a stump (very very simple tree)
- But simply using the package is too easy.
- We will do it a hard way.

What to boost ?

- Theoretically, you can boost anything.
- Yeh, anything but it needs some mod.
- For example,
  - SVM: by putting ‘C’ parameters on each of the slack variables and adjusting ‘C’ parameters according to D(i)
  - Logistic regression: by multiplying the likelihood term by D(i)

Let's boost LR

- First, we need to rewrite the objective function by including the D(i) terms.
- Next, find the first order partial derivatives
  - And also the Hessian.
- You are now ready to boost LR.
- Remember that LR is pretty stable classifier, try to shake it a bit to get diverse set of LRs.

AdaBoost Algorithm

Data: $S = \{x_i, y_i\}_{i=1}^{N}$, $x_i \in X$ and $y_i \in \{-1, 1\}$

initialization: uniform weight for initial data $D_1(i) = \frac{1}{N}$;

for $t = 1 \ldots T$ **do**

  Learn a classifier $h_t : X \to \{-1, 1\}$ that minimises training error, $\epsilon_j = \sum_{i=1}^{N} D_t(i)[y_i \neq h_j(x_i)]$;

  if $\epsilon_t > 0.5$ **then**

    STOP;

  else

    Set $\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t}$;

    Reweighting by $D_{t+1}(i) = \frac{D_t(i) \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$;

    (Z_t is a normaliser making $\sum_{i=1}^{N} D_{t+1}(i) = 1$);

  end

end

Result: $f_{ada}(x) = \sum_{t=1}^{T} \alpha_t h_t(x)$

Exercise 4

- Implement AdaBoost and try to boost logistic regression classifiers.

Classifier comparison

- Usually done using 5 or 10 fold cross validation
  - Divide data into 10 parts
  - Hold out 1 part at a time for testing
  - Final accuracy is the average of 10 (partial) accuracies
- Report the average and the standard deviations, for example
  - LR: 97.35 +/- 3.00 , SVM: 98.27 +/- 4.22

Is the difference significant ?

- Report the average and the standard deviations, for example
  - LR: 97.35 +/- 3.00 , SVM: 98.27 +/- 4.22
- Perform statistical test
  - People mostly use Wilcoxon ranksum test at 0.05 level of significant.
  - Consult the 1st Julia Slides for more example on statistical testing package.

Thank you !!

Work hard, play harder !