hw1_Yihui He

Tuesday, April 5, 2016 1:43 PM

#1

Pros:

1. Simple, easy to understand.
2. Fast when classifying, output can be determined directly.
3. It can deal with non-linear pattern(like xor).
4. It will be accurate when there is enough training samples.

Cons:

1. Space complexity is really high when there are multiple inputs, so it need huge space to store all pairs.
2. Can't deal with continuous input variable.
3. Not accurate when training samples are not enough.

#2

(a) San Francisco Crime Classification
Given time and location, Predict the category of crimes that occurred in the city by the bay. They're also encouraging kagglers to explore the dataset visually.

(b) incidents derived from SFPD Crime Incident Reporting system.

(c) Submissions are evaluated using the multi-class logarithmic loss. For each incident, you must submit a set of predicted probabilities (one for every class):

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}),$$

N is the number of cases, M is number of labels, y is label, p is predicted probabilities.

(d) The resolutions is related to incidents' categories.
Most incidents happened at night.

(e) Supervised learning, because all training data are labeled.

(f) Crime incidents data is given as follow:

- **Dates** - timestamp of the crime incident
- **Category** - category of the crime incident (only in train.csv). **This is the target variable you are going to predict.**
- **Descript** - detailed description of the crime incident (only in train.csv)
- **DayOfWeek** - the day of the week
- **PdDistrict** - name of the Police Department District
- **Resolution** - how the crime incident was resolved (only in train.csv)
- **Address** - the approximate street address of the crime incident
- **X** - Longitude
- **Y** - Latitude

#3

(a) Regression. More precisely, the task is predict numerical value, the experience are patients' health related data, and performance is measured by prediction is correct or not.
This is regression, because output is continuous numerical value.

(b) Predictive task.
We are asked to predict the LDL cholesterol level.

(c) Geometric model.

According to taxonomy in textbook, we need grading a lot, so we need geometric model.

(d) grading model.
Because our output is continuous numerical value. We need grading a lot.

(e) LDL cholesterol level.

(f) LDL cholesterol level. the same as label space.

#4

(a) $d_1=||x1-x2||_1=10.8$

(b) $d_2=||x1-x2||_2=5.8240879114244146$

(c) $d_{10}=||x1-x2||_{10}=4.2482768823608614$

(d) they will not change.
$d_k=\sum|(x1_k+v_k)-(x2_k-v_k)|^k=\sum|x1_k-x2_k|^k$

(e) all of them will change, unless k==1
$\sum|k*x1_n-k*x2_n|^n=|k|^n\sum|x1_n-x2_n|^n$

#5

(a)

	165B			basket	waving	
	small	medium	large	small	medium	large
A	0	0.185185	0.581395	0.333333	0.571429	0.857143
B	0.166667	0.296296	0.27907	0.333333	0.285714	0.142857
C	0.166667	0.37037	0.116279	0.333333	0.142857	0
D	0.333333	0.148148	0.023256	0	0	0
F	0.333333	0	0	0	0	0

(b)

	small	medium	large
A	0.05	0.125	0.275
B	0.075	0.09	0.085
C	0.075	0.075	0.025
D	0.05	0.02	0.005
F	0.05	0	0

(c)

small	medium	large
0.30	0.310	0.390

(d)

P(grade=A | class=165B)=0.3

P(grade=A | class=basketwaving)=0.6

#6

(a)

	predict face	predict non-face	
actual face	425	75	500
actual non-face	125	375	500
	550	450	1000

(b) 25%
(c) 15%
(d) 20%
(e) 77.27 %
(f) 80%

#7

(a) 2
(b) a line.
After dimensionality reduction, we have 2 dimension. So the hyperplane will be one dimension which is a line.
(c) To address this question, I perform a linear perceptron on dataset, because perceptron is guaranteed to find a solution with 100% accurate if the data is linearly separable.
Implementation is as follow:

```
x=np.array([[1.8,4,5.6],
            [4.8,7,8.6],
            [1.2,3.4,5.0],
            [4.5,6.7,8.3],
            [1.5,3.7,5.3],
            [4.2,6.4,8]])
y=np.array([1,-1,1,-1,-1,-1]) #A=1,B=-1
w=np.array([.0,.0,.0])
b=.0#homogeneous
n=0.01
maxiter = 10000
ite=0
converge=False
while converge==False:
    if ite==10000:
        break
    ite+=1
    converge=True
    for i in range(len(y)):
        if y[i]*(x[i].dot(w)+b)<=0:
            w+=n*y[i]*x[i]
            b+=n*y[i]
            converge=False
print "accuracy =",sum(y*(sum((w*x).T)+b)>0)/float(len(y))

the output is:
accuracy = 0.666666666667
```

So this dataset is not linearly separable