**Отчет**

**Муратова Анна, НОД, ИССА**

Разработанная программа выполняет:

1) исследование зависимости точности классификации от значения порога (превышения отношения количества примеров одного класса к другому) и нахождение оптимального порога для достижения лучшей точности классификации;

2) исследование зависимости точности классификации от объема тренировочных и тестовых данных (количества файлов).

**Описание программы “test\_impl”**

На вход программы подается обучающая выборка и разделяется на две части – с положительным целевым признаком и отрицательным. Далее открывается тестовая выборка, для которой надо определить целевой признак.

Для классификации объектов использовались формулы и . Если , то значение классифицируется положительно, если , то значение классифицируется отрицательно, иначе неопределенно. Здесь – значение порога.

Сравнивая каждое из значений в тестовой выборке со значениями из обучающей, подсчитывается количество TP, FP, TN, FN и Contradictory.

TP – правильные положительные, если мы предсказали положительный класс и реальное значение класса положительное (причем отношение положительных к отрицательным превышает порог)

FP – неправильные положительные, если мы предсказали положительный класс, а реальное значение класса отрицательное (причем отношение положительных к отрицательным превышает порог)

TN – правильные отрицательные, если мы предсказали отрицательный класс и реальное значение класса отрицательное (причем отношение отрицательных к положительным превышает порог)

FN – неправильные отрицательные, если мы предсказали отрицательный класс, а реальное значение класса положительное (причем отношение отрицательных к положительным превышает порог)

Contradictory – значения, не превышающие порог

**Описание программы “MyExecF”**

В программе задаются следующие параметры:

- количество файлов тренировочных и тестовых данных;

- интервал значений порога, на котором будет исследоваться зависимость точности от порога и нахождение оптимального порога.

Программа в цикле изменяет значение порога на заданном интервале с шагом 0,1 и вызывает программу “test\_impl”, которая считает количество TP, FP, TN, FN, Contradictory, а также точность для каждого значения порога для заданного количества файлов данных. При переборе вариантов выбирается значение лучшего порога, при котором точность максимальна.

Во второй части программы, при найденном значении лучшего порога, подсчитывается количество TP, FP, TN, FN, Contradictory и Accuracy для одного файла, двух файлов и так далее до заданного количества файлов данных. Таким образом исследуется зависимость точности от количества данных и находится при каком количестве данных (файлов) точность максимальна.

В результате программа находит лучшее значение порога и соответствующую ему точность. Также она выдает наиболее оптимальное количество файлов с количеством примеров и максимальной точностью.

Например, при значениях порога [0.5; 1.5] и количестве файлов 10, лучший порог получился 1.0 с точностью 0.921. Однако при расчете точности по переменному количеству файлов, лучший результат получился не для 10, как можно было бы предположить, а для 4-х файлов с 373 примерами. Точность в данном случае составила 0.919. Значение точности при одних и тех же данных варьируется при каждом запуске программы из-за применения функции random в программе test\_impl.

**Код программы “test\_impl”**

import sys

import random

import copy

attrib\_names = [

'top-left-square',

'top-middle-square',

'top-right-square',

'middle-left-square',

'middle-middle-square',

'middle-right-square',

'bottom-left-square',

'bottom-middle-square',

'bottom-right-square',

'class'

]

def make\_intent(example):

global attrib\_names

return set([i+':'+str(k) for i, k in zip(attrib\_names, example)])

cv\_res = {

"positive\_positive": 0,

"positive\_negative": 0,

"negative\_positive": 0,

"negative\_negative": 0,

"contradictory": 0,

"total": 0,

}

def check\_intersect(context\_plus, context\_minus, example, num\_sub=1):

global cv\_res

pos = 0

neg = 0

intent = make\_intent(example)

for i in xrange(num\_sub):

t = set(random.sample(example, random.randrange(len(intent))))

for j in context\_plus:

if t.issubset(j):

pos += len(t)

for k in context\_minus:

if t.issubset(k):

neg += len(t)

def score(pos, neg):

return pos \* 1. / (neg + 1)

#threshold = 1.1

if score(pos, neg) > threshold:

if example[-1] == 'positive':

cv\_res['positive\_positive'] += 1

else:

cv\_res['negative\_positive'] += 1

elif score(neg, pos) > threshold:

if example[-1] == 'positive':

cv\_res['positive\_negative'] += 1

else:

cv\_res['negative\_negative'] += 1

else:

cv\_res['contradictory'] += 1

def check\_hypothesis(context\_plus, context\_minus, example):

global cv\_res

eintent = make\_intent(example)

big\_context = context\_plus + context\_minus

labels = {}

for e in big\_context:

ei = make\_intent(e)

candidate\_intent = ei & eintent

if not candidate\_intent:

continue

closure = [make\_intent(i) for i in big\_context

if make\_intent(i).issuperset(candidate\_intent)]

res = reduce(lambda x, y: x & y if x & y else x | y, closure)

for cs in ['positive', 'negative']:

if 'class:' + cs in res:

labels[cs] = True

labels[cs + '\_res'] = candidate\_intent

if labels.get("positive", False) and labels.get("negative", False):

cv\_res["contradictory"] += 1

return

if example[-1] == "positive" and labels.get("positive", False):

cv\_res["positive\_positive"] += 1

if example[-1] == "negative" and labels.get("positive", False):

cv\_res["negative\_positive"] += 1

if example[-1] == "positive" and labels.get("negative", False):

cv\_res["positive\_negative"] += 1

if example[-1] == "negative" and labels.get("negative", False):

cv\_res["negative\_negative"] += 1

# Get data from train and test files

#max\_index = sys.argv[1]

for index in xrange(1, int(max\_index)):

index = str(index)

q = open("train" + index + ".csv", "r")

train = [a.strip().split(",") for a in q]

plus = [a for a in train if a[-1] == "positive"]

minus = [a for a in train if a[-1] == "negative"]

q.close()

w = open("test" + index + ".csv", "r")

unknown = [a.strip().split(",") for a in w]

w.close()

for elem in unknown:

cv\_res['total'] += 1

check\_intersect(plus, minus, elem, len(elem) / 2)

cv\_res\_p = copy.copy(cv\_res)

total = cv\_res\_p["total"]

for k, v in cv\_res\_p.iteritems():

cv\_res\_p[k] = v \* 1. / total # part of 1.0

print "Number of datasets done = %s" % index

#print cv\_res

**Код программы “MyExecF”**

# Test of Implication Test by Threshold

import sys

#max\_index = sys.argv[1]

max\_files = 11

ithr\_min = 5 # min Threshold \* 10

ithr\_max = 16 # max Threshold \* 10

max\_index = max\_files

accuracy = {} # Accuracy for different Threshold

accuracyf = {} # Accuracy for different number of files

acc\_max = 0 # best Accuracy for different Threshold

acc\_maxf = 0 # best Accuracy for different number of files

ithr\_best = ithr\_min # index of best Accuracy for different Threshold

imax\_best = 1 # index of best Accuracy for different number of files

totalf = {} # total number of examples through all files

for ithreshold in xrange(ithr\_min, ithr\_max, 1):

threshold = 0.1\*ithreshold

execfile("test\_impl.py")

accuracy[ithreshold] = cv\_res\_p["positive\_positive"] + cv\_res\_p["negative\_negative"] # Accuracy

if accuracy[ithreshold] > acc\_max :

acc\_max = accuracy[ithreshold]

ithr\_best = ithreshold

print 'Threshold =', threshold, '\tAccuracy =', accuracy[ithreshold]

print 'True Positive :\t', cv\_res["positive\_positive"], '\t', cv\_res\_p["positive\_positive"]

print 'True Negative :\t', cv\_res["negative\_negative"], '\t', cv\_res\_p["negative\_negative"]

print 'False Positive :\t', cv\_res["negative\_positive"], '\t', cv\_res\_p["negative\_positive"]

print 'False Negative :\t', cv\_res["positive\_negative"], '\t', cv\_res\_p["positive\_negative"]

print 'Contradictory :\t', cv\_res["contradictory"], '\t', cv\_res\_p["contradictory"]

print 'Total :\t', cv\_res["total"], '\t', cv\_res\_p["total"]

print '========================================================'

print '========================================================'

for max\_index in xrange(2, int(max\_files)+1):

threshold = 0.1\*ithr\_best # Best Threshold will be used

execfile("test\_impl.py")

accuracyf[max\_index] = cv\_res\_p["positive\_positive"] + cv\_res\_p["negative\_negative"] # Accuracy

if accuracyf[max\_index] > acc\_maxf :

acc\_maxf = accuracyf[max\_index]

imax\_best = max\_index

totalf[max\_index] = cv\_res["total"]

print 'Number of Files =', max\_index-1, '\tAccuracy =', accuracyf[max\_index]

print 'True Positive :\t', cv\_res["positive\_positive"], '\t', cv\_res\_p["positive\_positive"]

print 'True Negative :\t', cv\_res["negative\_negative"], '\t', cv\_res\_p["negative\_negative"]

print 'False Positive :\t', cv\_res["negative\_positive"], '\t', cv\_res\_p["negative\_positive"]

print 'False Negative :\t', cv\_res["positive\_negative"], '\t', cv\_res\_p["positive\_negative"]

print 'Contradictory :\t', cv\_res["contradictory"], '\t', cv\_res\_p["contradictory"]

print 'Total :\t', cv\_res["total"], '\t', cv\_res\_p["total"]

print '========================================================'

print "Threshold \tAccuracy"

print "--------- \t--------------"

for ithreshold in xrange(ithr\_min, ithr\_max, 1):

if ithreshold == ithr\_best:

bestsign = "!!! Best !!!"

else:

bestsign = " "

print 0.1\*ithreshold, "\t\t", accuracy[ithreshold], "\t", bestsign

print '========================================================'

print "Files \t\tExamples \tAccuracy"

print "-------\t\t-------- \t--------------"

for max\_index in xrange(2, int(max\_files)+1):

if max\_index == imax\_best:

bestsign = "!!! Best !!!"

else:

bestsign = " "

print max\_index-1, "\t\t", totalf[max\_index], "\t\t", accuracyf[max\_index], "\t", bestsign

print '========================================================'

**Результат программы (порог [0.5; 1.5], файлов 10)**

Python 2.7.9 (default, Dec 10 2014, 12:24:55) [MSC v.1500 32 bit (Intel)] on win32

Type "copyright", "credits" or "license()" for more information.

>>> ================================ RESTART ================================

>>>

Number of datasets done = 10

Threshold = 0.5 Accuracy = 0.779958677686

True Positive : 626 0.646694214876

True Negative : 129 0.13326446281

False Positive : 203 0.209710743802

False Negative : 0 0.0

Contradictory : 10 0.0103305785124

Total : 968 1.0

========================================================

Number of datasets done = 10

Threshold = 0.6 Accuracy = 0.815082644628

True Positive : 626 0.646694214876

True Negative : 163 0.168388429752

False Positive : 169 0.17458677686

False Negative : 0 0.0

Contradictory : 10 0.0103305785124

Total : 968 1.0

========================================================

Number of datasets done = 10

Threshold = 0.7 Accuracy = 0.807851239669

True Positive : 626 0.646694214876

True Negative : 156 0.161157024793

False Positive : 176 0.181818181818

False Negative : 0 0.0

Contradictory : 10 0.0103305785124

Total : 968 1.0

========================================================

Number of datasets done = 10

Threshold = 0.8 Accuracy = 0.848140495868

True Positive : 626 0.646694214876

True Negative : 195 0.201446280992

False Positive : 137 0.14152892562

False Negative : 0 0.0

Contradictory : 10 0.0103305785124

Total : 968 1.0

========================================================

Number of datasets done = 10

Threshold = 0.9 Accuracy = 0.884297520661

True Positive : 626 0.646694214876

True Negative : 230 0.237603305785

False Positive : 102 0.105371900826

False Negative : 0 0.0

Contradictory : 10 0.0103305785124

Total : 968 1.0

========================================================

Number of datasets done = 10

Threshold = 1.0 Accuracy = 0.921487603306

True Positive : 626 0.646694214876

True Negative : 266 0.27479338843

False Positive : 66 0.0681818181818

False Negative : 0 0.0

Contradictory : 10 0.0103305785124

Total : 968 1.0

========================================================

Number of datasets done = 10

Threshold = 1.1 Accuracy = 0.877066115702

True Positive : 626 0.646694214876

True Negative : 223 0.230371900826

False Positive : 81 0.0836776859504

False Negative : 0 0.0

Contradictory : 38 0.0392561983471

Total : 968 1.0

========================================================

Number of datasets done = 10

Threshold = 1.2 Accuracy = 0.851239669421

True Positive : 626 0.646694214876

True Negative : 198 0.204545454545

False Positive : 64 0.0661157024793

False Negative : 0 0.0

Contradictory : 80 0.0826446280992

Total : 968 1.0

========================================================

Number of datasets done = 10

Threshold = 1.3 Accuracy = 0.827479338843

True Positive : 626 0.646694214876

True Negative : 175 0.180785123967

False Positive : 35 0.0361570247934

False Negative : 0 0.0

Contradictory : 132 0.136363636364

Total : 968 1.0

========================================================

Number of datasets done = 10

Threshold = 1.4 Accuracy = 0.818181818182

True Positive : 626 0.646694214876

True Negative : 166 0.171487603306

False Positive : 22 0.0227272727273

False Negative : 0 0.0

Contradictory : 154 0.159090909091

Total : 968 1.0

========================================================

Number of datasets done = 10

Threshold = 1.5 Accuracy = 0.828512396694

True Positive : 626 0.646694214876

True Negative : 176 0.181818181818

False Positive : 27 0.0278925619835

False Negative : 0 0.0

Contradictory : 139 0.143595041322

Total : 968 1.0

========================================================

========================================================

Number of datasets done = 1

Number of Files = 1 Accuracy = 0.893617021277

True Positive : 61 0.648936170213

True Negative : 23 0.244680851064

False Positive : 9 0.0957446808511

False Negative : 0 0.0

Contradictory : 1 0.0106382978723

Total : 94 1.0

========================================================

Number of datasets done = 2

Number of Files = 2 Accuracy = 0.873626373626

True Positive : 112 0.615384615385

True Negative : 47 0.258241758242

False Positive : 21 0.115384615385

False Negative : 0 0.0

Contradictory : 2 0.010989010989

Total : 182 1.0

========================================================

Number of datasets done = 3

Number of Files = 3 Accuracy = 0.91519434629

True Positive : 177 0.625441696113

True Negative : 82 0.289752650177

False Positive : 21 0.0742049469965

False Negative : 0 0.0

Contradictory : 3 0.0106007067138

Total : 283 1.0

========================================================

Number of datasets done = 4

Number of Files = 4 Accuracy = 0.919571045576

True Positive : 236 0.632707774799

True Negative : 107 0.286863270777

False Positive : 26 0.0697050938338

False Negative : 0 0.0

Contradictory : 4 0.0107238605898

Total : 373 1.0

========================================================

Number of datasets done = 5

Number of Files = 5 Accuracy = 0.88120950324

True Positive : 298 0.643628509719

True Negative : 110 0.237580993521

False Positive : 50 0.107991360691

False Negative : 0 0.0

Contradictory : 5 0.0107991360691

Total : 463 1.0

========================================================

Number of datasets done = 6

Number of Files = 6 Accuracy = 0.888888888889

True Positive : 354 0.644808743169

True Negative : 134 0.244080145719

False Positive : 55 0.100182149362

False Negative : 0 0.0

Contradictory : 6 0.0109289617486

Total : 549 1.0

========================================================

Number of datasets done = 7

Number of Files = 7 Accuracy = 0.896084337349

True Positive : 424 0.638554216867

True Negative : 171 0.257530120482

False Positive : 62 0.0933734939759

False Negative : 0 0.0

Contradictory : 7 0.0105421686747

Total : 664 1.0

========================================================

Number of datasets done = 8

Number of Files = 8 Accuracy = 0.895077720207

True Positive : 497 0.64378238342

True Negative : 194 0.251295336788

False Positive : 73 0.0945595854922

False Negative : 0 0.0

Contradictory : 8 0.0103626943005

Total : 772 1.0

========================================================

Number of datasets done = 9

Number of Files = 9 Accuracy = 0.900684931507

True Positive : 567 0.647260273973

True Negative : 222 0.253424657534

False Positive : 78 0.0890410958904

False Negative : 0 0.0

Contradictory : 9 0.0102739726027

Total : 876 1.0

========================================================

Number of datasets done = 10

Number of Files = 10 Accuracy = 0.89152892562

True Positive : 626 0.646694214876

True Negative : 237 0.244834710744

False Positive : 95 0.0981404958678

False Negative : 0 0.0

Contradictory : 10 0.0103305785124

Total : 968 1.0

========================================================

Threshold Accuracy

--------- --------------

0.5 0.779958677686

0.6 0.815082644628

0.7 0.807851239669

0.8 0.848140495868

0.9 0.884297520661

1.0 0.921487603306 !!! Best !!!

1.1 0.877066115702

1.2 0.851239669421

1.3 0.827479338843

1.4 0.818181818182

1.5 0.828512396694

========================================================

Files Examples Accuracy

------- -------- --------------

1 94 0.893617021277

2 182 0.873626373626

3 283 0.91519434629

4 373 0.919571045576 !!! Best !!!

5 463 0.88120950324

6 549 0.888888888889

7 664 0.896084337349

8 772 0.895077720207

9 876 0.900684931507

10 968 0.89152892562

========================================================

>>>