Old Dominion Univeristy

CS 495: Introduction to Web Science
Instructor: Michael L. Nelson, Ph.D
Fall 2014 4:20pm - 7:10pm R, ECSB 2120

Assignment # 10

George C. Micros UIN: 00757376

Honor Pledge

I pledge to support the Honor System of Old Dominion University. I will refrain from any form of academic dishonesty or deception, such as cheating or plagIiarism. I am aware that as a member of the academic community it is my responsibility to turn in all suspected violations of the Honor Code. I will report to a hearing if summoned.

Signed _____
December 11, 2014

George C. Micros

Written Assignment 10

Fall 2014
CS 495: Introduction to Web Science
Dr. Michael Nelson

December 11, 2014

Contents

Chapter 1
Written Assignment 10

1.1 Question 1

1.1.1 The Question

Choose a blog or a newsfeed (or something similar as long as it has an Atom or RSS feed). It should be on a topic or topics of which you are qualified to provide classification training data. In other words, choose something that you enjoy and are knowledgable of. Find a feed with at least 100 entries.
Create between four and eight different categories for the entries in the feed:
examples:

- work, class, family, news, deals
- liberal, conservative, moderate, libertarian
- sports, local, financial, national, international, entertainment
- metal, electronic, ambient, folk, hip-hop, pop

Download and process the pages of the feed as per the week 12 class slides.

1.1.2 The Answer

A blog from blogspot was chosen based on the topic and how well it could be described with categories. The topic selected is soccer. The categories are: Spain, England, World, Americas, Germany, Africa, because these are the main themes that arise in the posts. A script was written that uses cURL to make a request of the atom feed page with an additional query of 100 results. This is saved as an .xml file. The classification was done manually and written in "class.txt".

```
1 #! /bin/bash

3 curl -L "http://soccerphile.blogspot.com/feeds/posts/default?max-results=100" > feed.xml
```

Listing 1.1: Bash script to retrieve Atom feed from blog

1.2 Question 2

1.2.1 The Question

Manually classify the first 50 entries, and then classify (using the fisher classifier) the remaining 50 entries. Report the cprob() values for the 50 titles as well. From the title or entry itself, specify the 1-, 2-, or 3-gram that you used for the string to classify. Do not repeat strings; you will have 50 unique strings. For example, in these titles the string used is marked with *s:
Rachel Goswell - "Waves Are Universal" (LP Review) The *Naked and Famous* - "Passive Me, Aggressive You" (LP Review) *Negativland* - "Live at Lewis's, Norfolk VA, November 21, 1992" (concert) Negativland - "*U2*" (LP Review)
Note how "Negativland" is not repeated as a classification string.
Create a table with the title, the string used for classification, cprob(), predicted category, and actual category.

1.2.2 The Answer

The list of entries was classified manually into a text file that was used as input to the code. The code primarily consisted of functions borrowed from the book code and modified to the project. The classifer and feedfilter modules were also imported for their other functionality.

```
#! /usr/bin/python
import feedparser
import feedfilter
import docclass
import random
import re
cl=docclass.fisherclassifier(docclass.getwords)
cl.setdb('soccerphile.db')

feed = "feed.xml"
f=feedparser.parse(feed)
g = open("class.txt","r")
next(g)
cat = []
for line in g:
  tmp = line.split(' ')[1].strip('\n')
  cat.append(tmp)

num = 50

def q1Train(f,cat,num,classifier):
  # Get feed entries and loop over them
  print "--- TRAINING ---"
  for entry in f['entries']:
    i = f.entries.index(entry)
    if( i < num):
      print
      print '-----'
      print 'Title:      '+entry['title'].encode('utf-8')
      fulltext='%s\n%s' % (entry['title'],entry['summary'][0])
      print i
      classifier.train(fulltext, cat[i])

  for i in classifier.categories():
    classifier.setminimum(i,.16)

def q1Test(f,cat,num,classifier):
  print "--- TESTING ---"
  h = open("table.txt", "w")
  h.write("title, string, cprob, pred, act\n")

  for entry in f['entries']:
    i = f.entries.index(entry)
    if( i >= num):
      print
      print '-----'
      print 'Title:      '+entry['title'].encode('utf-8')
```

```
        fulltext='%s\n%s' % (entry['title'],entry['summary'])
49      print i
        a = feedfilter.entryfeatures(entry)
51      b = random.randint(0,len(a)-1)
        c = ' '.join(a)
53      #print a
        fulltext='%s\n%s' % (entry['title'],entry['summary'][0])
55      # cls = classifier.classify(a.items()[b][0])
        # cls = classifier.classify(fulltext)
57      cls = classifier.classify(fulltext)

59      f.entries[i].pred = cls
        f.entries[i].cls = cat[i]
61      # cls=raw_input('Enter category: ')
        # cprob = classifier.fisherprob(entry['title'], cls)
63      cprob = classifier.fisherprob(a.items()[b][0], cls)
        cprob = classifier.fisherprob(c, cls)
65      print "Guess: "+str(cls)
        print "Actual: "+str(cat[i])
67      print "C Prob: "+str(cprob)

69      fout = '%s, %s, %f, %s, %s\n' % (re.sub(r'[^\x00-\x7F]+', '', entry['title'][0:30]), a.
    items()[b][0], cprob, cls, cat[i])
        h.write(fout)
71  def fmeasure(f, num):
    tp, fp, fn = 0, 0, 0
73  for i in range(num, 100):
      if(f.entries[i].pred == f.entries[i].cls):
75      tp +=1
      if(f.entries[i].pred != f.entries[i].cls):
77      fp +=1
      if(not f.entries[i].pred):
79      fn +=1

81  prec = float(tp)/(tp + fp)
    recl = float(tp)/(tp +fn)
83
    fmeasure = 2*(prec*recl)/(prec+recl)
85  print prec, recl
    print fmeasure
87
q1Train(f, cat, num, cl)
89  q1Test(f, cat, num, cl)
fmeasure(f,num)
```

Listing 1.2: Python script for training and tesing classfier

The classification using random 1-grams from the title and summary produced very poor results. This is most likely to the fact that the classification is content based and key terms, which would be very effective features, are very sparse. The terms of the classification could be manually selected to convey specific significance and be more appropriate features fo the classification. However, aside from tedious this is also counterproductive. The performance of a classifier should be evaluated under real-world conditions with imperfect data and noise. Therefore, selecting the key terms for classification would only bias the results.

Using only random 1-gram was meaningless and inefficient. There are very few key terms that can differentiate classes in this data, i.e. cup names, player names, football club names. In an attempt to increase accuracy and performance the process was repeated using the title of each entry for the classification. Both results are diaplyed below

In thi situation there were 6 classes making chance accuracy approximately 0.1666. This is the probability of correct classification based on random chance. The classification selects the largest class probability, which is the fischer's probability of belonging to a class. However, because the largest one is always selected there is always aclassification, even with a very small fischer's probability. These types of classifications are the equvalant of guessing and are useless. A restriction was placed in the form of a minimum fischer's probability requirement. The minimum fisher's probability was chosen heuristically at value of p = .16 by using various values and selecting the one that produced the highest f-measure for the classifier

Table 1.1: Results of classification based on random 1-gram

title	string	cprob	pred	act
Belo Horizonte	once	0.500000	Spain	World
Departure for Brazil	400	0.500000	Spain	World
Introduction to the 2014 World	travels	0.500000	Spain	World
Fifa World Rankings June 2014	separator	0.500000	Spain	World
Nigeria share goals with Scots	brown	0.500000	Spain	World
International Friendly: Scotla	off	0.500000	Spain	World
Falcao & Suarez risk World Cup	both	0.500000	Spain	World
Donovan dropped from US squad	bold	0.500000	Spain	Americas
This is the year for Brazil	muscular	0.500000	Spain	Americas
Super Eagles are ready for tak	cut	0.500000	Spain	Africa
Scudamore keeps his job but we	from	0.500000	Spain	England
Cantona aims kick at Platini	this	0.500000	Spain	England
Sepp comes clean on Qatar	suggesting	0.500000	Spain	World
Thiago blow for Spain	align	0.500000	Spain	Spain
C'est la vie says Nasri	celebrity	0.500000	Spain	England
World Cup squad announcements	arm	0.500000	Spain	World
Now if my name were Roy...	home	0.500000	Spain	England
George's Premiership Predictio	leg	0.500000	Spain	England
Fifa World Rankings May 2014	come	0.500000	Spain	World
George's Premiership Predictio	land	0.500000	Spain	England
AFC Champions League Rotation	season	0.500000	Spain	World
Champions League Bayern vs Rea	crowned	0.500000	Spain	World
George's Premiership Predictio	sees	0.500000	Spain	England
EURO 2016 Finals schedule anno	championship	0.500000	Spain	World
Scottish Premier League News A	field	0.500000	Spain	England
David Moyes Sacked	won	0.500000	Spain	England
George's Premiership Predictio	that	0.500000	Spain	England
What is wrong with Bara?	germain	0.500000	Spain	Spain
A-League Finals 2014	screen	0.500000	Spain	World
The Meaning of Hillsborough 25	ignorance	0.500000	Spain	England
History Tells Us That Brazil M	blind	0.500000	Spain	Americas
George's Premiership Predictio	former	0.500000	Spain	England
Fifa World Rankings April 2014	right	0.500000	Spain	World
A Race that never ends	their	0.500000	Spain	England
40 a goal adds insult to Spur	wake	0.500000	Spain	England
Man U & Real top replica sales	jerseys	0.500000	Spain	England
Weekly Football News April 7 2	football	0.500000	Spain	World
George's Premiership Predictio	must	0.500000	Spain	England
Is the J-League Too Big?	number	0.500000	Spain	England
George's Premiership Predictio	improved	0.500000	Spain	England
Weekly Football News March 25	front	0.500000	Spain	World
Scottish Premier League News M	dillon	0.500000	Spain	England
George's Premiership Predictio	shown	0.500000	Spain	England
George's Premiership Predictio	struggling	0.500000	Spain	England
England & Italy victims of Eur	injections	0.500000	Spain	World
Fifa World Rankings March 2014	under	0.500000	Spain	World
Despite Featuring Many Top Tea	right	0.741745	England	World
Revenge served cold on Pohang	left	0.500000	Spain	England
Weekly Football News March 11	went	0.500000	Spain	World
George's Premiership Predictio	run	0.500000	Spain	England

Table 1.2: Results of classification based on title string

title	string	cprob	pred	act
Belo Horizonte		0.999543	World	World
Departure for Brazil		0.005880	None	World
Introduction to the 2014 World		0.570638	World	World
Fifa World Rankings June 2014		0.940018	World	World
Nigeria share goals with Scots		0.731898	Spain	World
International Friendly: Scotla		0.839642	World	World
Falcao & Suarez risk World Cup		0.682801	World	World
Donovan dropped from US squad		0.007454	None	Americas
This is the year for Brazil		0.000496	None	Americas
Super Eagles are ready for tak		0.355228	World	Africa
Scudamore keeps his job but we		0.936780	World	England
Cantona aims kick at Platini		0.698030	Spain	England
Sepp comes clean on Qatar		0.698030	Spain	World
Thiago blow for Spain		0.005529	None	Spain
C'est la vie says Nasri		0.698030	Spain	England
World Cup squad announcements		0.000048	None	World
Now if my name were Roy...		0.698030	Spain	England
George's Premiership Predictio		0.996521	England	England
Fifa World Rankings May 2014		0.940018	World	World
George's Premiership Predictio		0.996521	England	England
AFC Champions League Rotation		0.839642	England	World
Champions League Bayern vs Rea		0.849562	England	World
George's Premiership Predictio		0.996521	England	England
EURO 2016 Finals schedule anno		0.936879	World	World
Scottish Premier League News A		0.462349	England	England
David Moyes Sacked		0.655185	Spain	England
George's Premiership Predictio		0.996521	England	England
What is wrong with Bara?		0.839642	Africa	Spain
A-League Finals 2014		0.919880	England	World
The Meaning of Hillsborough 25		0.759110	Germany	England
History Tells Us That Brazil M		0.645775	World	Americas
George's Premiership Predictio		0.994320	England	England
Fifa World Rankings April 2014		0.940018	World	World
A Race that never ends		0.839642	World	England
40 a goal adds insult to Spur		0.731898	Spain	England
Man U & Real top replica sales		0.731898	Spain	England
Weekly Football News April 7 2		0.003232	None	World
George's Premiership Predictio		0.994320	England	England
Is the J-League Too Big?		0.000207	None	England
George's Premiership Predictio		0.996521	England	England
Weekly Football News March 25		0.003232	None	World
Scottish Premier League News M		0.462349	England	England
George's Premiership Predictio		0.996521	England	England
George's Premiership Predictio		0.996521	England	England
England & Italy victims of Eur		0.398828	England	World
Fifa World Rankings March 2014		0.940018	World	World
Despite Featuring Many Top Tea		0.913444	World	World
Revenge served cold on Pohang		0.698030	Spain	England
Weekly Football News March 11		0.003232	None	World
George's Premiership Predictio		0.996521	England	England

1.3 Question 3

1.3.1 The Question

Assess the performance of your classifier in each of your categories by computing precision, recall, and F1. Note that the definitions of precisions and recall are slightly different in the context of classification; see:

`http://en.wikiedia.org/wiki/Precision_and_recall#Definition_.28classification_context.29`
and
`http://en.wikipedia.org/wiki/F1_score`

1.3.2 The Answer

In information retreival and other classification scenarios it is important to be able to assign a value to the performance of a techniques based on accuracy of the reulting queries. Being able to understand and measure the relevance of the results produced allows for a better understanding of the functionality of the classifier and how it fails in certain circumstances.

Precision measures how closely the retreived documents fit the query that was placed. Precision is the fraction of retrieved documents that are relevant. High precision produces a high proportion of relevant results.

$$\text{precision} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{retrieved documents}\}|}$$

Recall measures how effectively the relevant documents can be retreived. Recall is the fraction of the documents that are relevant to the query that are succesfully retrieved. This provides a counterbalance to precision. It is possible to produce documents that are all relevant to the query and therefore have high precision. However, the ability to retrieve all documents relevant from the corpus is recall. If strict restrictions are placed to provide a high precision, the recall may suffer in return. However, in the context of classification it is possible to train a classifier to produce both high recall and precision

$$\text{recall} = \frac{|\{\text{relevant documents}\} \cap \{\text{retrieved documents}\}|}{|\{\text{relevant documents}\}|}$$

The consideration of both precision and recall produces the F-score, or F-measure. The F-measure is a metric of a tests overll accuracy, factoring in both precision adn recall.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

In addition to these definitions precision and recall can be defined from the confusion matrix of the classifier as follows.

$$\text{Precision} = \frac{tp}{tp+fp} \text{ , Recall} = \frac{tp}{tp+fn}$$

In this situation true positive(tp) represent that posts that were classified correctly. False positive (fp) are the results that are misclassified. False negative (fn) are the results that were not classified at all due to their low probability of belonging in any class. This is expressed in the follwing function that computes precision, recall and the f-measure for the classifier.

```
def fmeasure(f, num):
  tp, fp, fn = 0, 0, 0
  for i in range(num, 100):
    if(f.entries[i].pred == f.entries[i].cls):
      tp +=1
    if(f.entries[i].pred != f.entries[i].cls):
      fp +=1
    if(not f.entries[i].pred):
      fn +=1

  prec = float(tp)/(tp + fp)
  recl = float(tp)/(tp +fn)

  fmeasure = 2*(prec*recl)/(prec+recl)
  print prec, recl
  print fmeasure
```

Listing 1.3: Python implementation of F-measure

As mentioned previously, two different classification procedures were chosen. One followed the assignment instructions using n-grams from the title and entry. This consistently produced low results as shown below

$$\text{Precision} = 0.08, \text{Recall} = 1.0$$
$$\text{F-measure} = 0.148148148148$$

It appeared that experimenting with other classification inputs the accuracy could be improved. The title of the posts presents a lot of information that is relavant to the class of the post. Using the entire title as an input the following results were measured.

$$\text{Precision} = 0.48, \text{Recall} = 0.857142857143$$
$$\text{F-measure} = 0.615384615385$$

This seems reasonable and intuitive in a sense. Expecting classification off a single word or two is very stringent. However, expanding the input features the possibilities can be narrowed down. In one of the attempts to select useful features the title and summary were both used are input string. This was unsuccessful as the probability of the entire string went to zero and all infomation was lost. Clearly feature size and performance do not have a linear relationship, but there is some region were increasing or decreasing the size of features used can produce a roughly linear response. This region is typically were the maxima of classifier performance if located. It takes some tinkering to fine tune it may never converge, given that input should be changing all the time.

1.4 Question 4

1.4.1 The Question

Redo questions 2 & 3, but with manually train 90 entries and then classify the remaining 10.
Then redo questions 2 & 3, but with the extensions on slide 26 and pp. 136–138. Fully discuss the changes you've made.
Which method (more training vs. better features) gave better improvement over your baseline? Why do you think that is?

1.4.2 The Answer

Applying the same classifier that was trained on 90 entries and tested on 10 produced the following results.
Training off 90 entries:

$$\text{Precision} = 0.6, \text{Recall} = 0.75$$
$$\text{F-measure} = 0.666666666667$$

The "entry features" function returned a list of terms, these terms were then made into a string and passed to the classifier for testing.

```
a = feedfilter.entryfeatures(entry)
      b = random.randint(0,len(a)-1)
      c = ' '.join(a)
      #print a
      fulltext='%s\n%s' % (entry['title'],entry['summary'][0])
      # cls = classifier.classify(a.items()[b][0])
      # cls = classifier.classify(fulltext)
      cls = classifier.classify(fulltext)
```

Listing 1.4: Python code using entry features

Using 50 entries and entryfeatures() vector:

$$\text{Precision} = 0.9, \text{Recall} = 0.957446808511$$
$$\text{F-measure} = 0.927835051546$$

It is clear from the results that using better features yield higher accuracy than more training. A large training set is important for a model to generalize well and perform accurately with other data. However, increasing the training data does not necessarily correlate with performance, because the training set may contain noisy data with redundant information and the classifier is trained on insufficient information. Howerver, carefully selecting features that are good predictors of the class will increase performance and allow a model to generalize better. An analogy is spending hours studying the same type of math problem and neglecting other types of problems, rather than looking a small sample of different problems.

References

1. Toby Segaran. *Programming collective intelligence: building smart web 2.0 applications.* " O'Reilly Media, Inc.", 2007.
2. Maja J Mataric. Designing emergent behaviors: From local interactions to collective intelligence. In *Proceedings of the Second International Conference on Simulation of Adaptive Behavior*, pages 432–441, 1993.
3. Eric Cai The Chemical Statistician. Exploratory data analysis: Combining histograms and density plots to examine the distribution of the ozone pollution data from new york in r. `http://www.r-bloggers.com/exploratory-data-analysis-combining-histograms-and-density-plots-to-examine-the-distribution-of-the-ozone-pollution-data-fr`