Hacettepe University
Department of Computer Science & Engineering

BBM 204 SOFTWARE LABORATORY II
EXPERIMENT 5

for section 1

Subject : String Processing & Data Mining

Submission Date : 29 May 2013

Deadline : 12 Jun 2013

Language/version : Java 7th SE

Advisor : R.A. Yasin Şahin, Dr. Erkut ERDEM

1. INTRODUCTION

Text Mining

Text Mining is a subtitle in data mining which extracts meaningful information from unstructured data. In our days, huge size of data comes from textual data. It is reported that, 80% of data is unstructured [1] and most of unstructured data is stored in textual format. All of the magazines, papers, e-mails, and search queries are pressed or stored in text. Not only are the text written in natural languages handled by text mining, but also discipline-aided languages such as protein interactions within bio-informatics profit are also handled by text mining. Because of these reasons, people studying data mining gravitates towards text mining inevitably.

Context based text classification is one of the problems in text mining. Classification tasks can be divided into three types: supervised classification where classification rules are given externally, unsupervised classification where the classification must be done without any background information, and semi-supervised classification, where some of the classes are labelled before.

Outbreak Detection

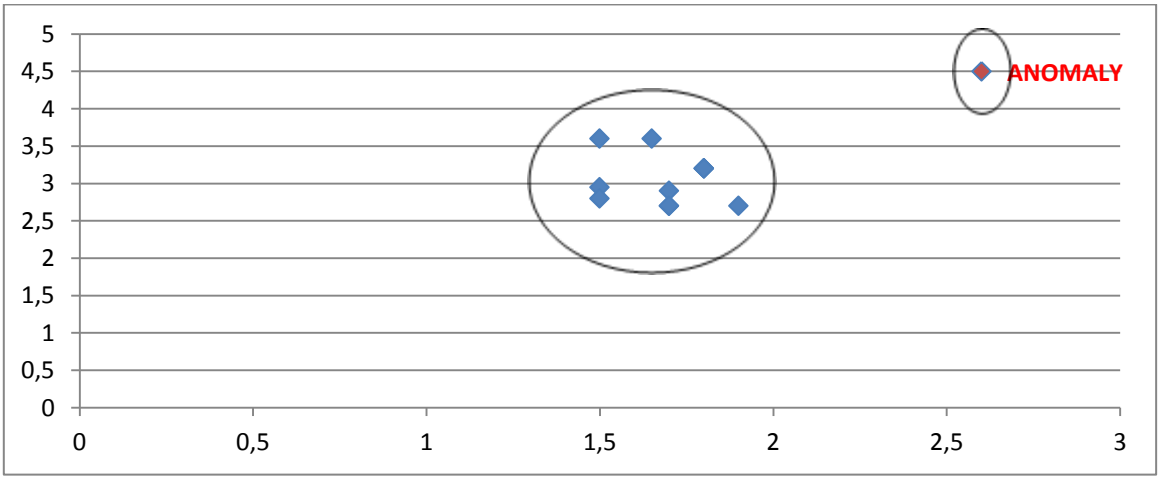

Figure 1 Anomaly

Anomaly detection is another subtitle in data mining. Anomaly is also referred to outlier [2] , intrusion, fraud, outbreak etc. Each of this words mean the same thing: aberrant behaviours, patterns placed into normal behaviours (see figure 1). Statisticians call it outlier, network security researchers call it intrusion or fraud for money works. Anomaly is called outbreak by the researchers who study on epidemiology. In other

words, outbreak is a term which is used in epidemiology to describe an occurrence of a disease greater than would otherwise be expected at a particular time and place [3]. Researchers propose several techniques to detect an outbreak. Each of these techniques is considered as a data mining method such as Statistical Process Control (SPC) charts, Machine Learning methods etc.

Shewhart Charts

Shewhart is the best known method for Statistical Process Control (SPC) and it is basis of SPC [4]. It is based on a manufacturing problem in 1920s. W.A. Shewhart proposed a statistical formula (eq.1.) to monitor the process of manufacturing to detect problems in the production quality. Shewhart method measures the shift score of the observation from an expected value.

$$S_t = \frac{y_t - \mu}{\sigma} \qquad \text{Eq.1.}$$

where S_t is the Shewhart score, y_t is the observation count at time t, μ is the expected value(i.e. mean) and σ is the standard deviation. In the context of epidemiology, Shewhart score has been used for the early detection of outbreaks in *biosurveillance* systems. If S_t is greater than the threshold then signal is raised about outbreak by the system. However many systems do not work in real time because of live dataset collecting constraints. An ordinary biosurveillance system collects the data at the end of a weekly, monthly or yearly period from data provider points.

2. EXPERIMENT

We search everything that we want to reach wherever we are. Therefore, we learn many things such as how to cook something that we want to eat, how to go to a place where we want to reach or how to solve pc problems that we encounter. Another thing that we search is about how to get rid of our diseases without consulting a doctor. We write down the indications into search box in order to get some helpful suggestions.

Everyday millions of users around the world search for health information online [5]. This information is used by Google to determine Influenza-like illness (ILI) risk map for the supported countries such as USA, Germany, Russia and many others.

In this experiment, you are supposed to design and implement a Public Health Surveillance System to monitor the past events by the query data provided by an online search engine. Assume that, a search engine that is preferred widely in the country, keeps logs of queries. Also assume that the search engine serves logs to the developers who work for public health researches. Your program is expected to monitor outbreak status of different locations day by day from past data.

Firstly, your program will get an input file that includes diseases and their symptoms. An illness (syndrome) can cause too many symptoms. For example, influenza causes headache, pyrexia, nasal flow etc. Moreover, a symptom can give clues for more than one disease. For example headache can be caused by influenza, migraine etc. As you can see, if a patient searches about headache, it doesn't match one disease smoothly. So, we need more then headache to achieve robust decision. In the real life, it is usually not possible to detect disease by only the given symptoms. However, we can classify the symptoms into diseases. For instance, if a enquiry contains *nasal flow* and *pyrexia* keywords, we can calculate the probability of the syndrome being to seasonal influenza. In order to handle this problem, you will use a similarity measurement method called Dice Coefficient. Dice coefficient is an equation to compare similarities of two samples.

$$dice = \frac{2x|A \cap B|}{|A| + |B|} \qquad \text{Eq.2.}$$

Sample Dice Coefficient similarity measurement:

Symptoms of an illness: *fever, headache, stomach ache*

Disease 1: ***stomach ache***, ***fever***, *nausea, asthenia (stomach ache and fever are common)*

Disease 2: ***headache***, *pruritus, stiff neck (headache is common)*

$$\text{dice1: } \frac{2x2}{3+4} = \frac{4}{7} \cong 0{,}57$$

$$\text{dice2: } \frac{2x1}{3+3} = \frac{2}{6} \cong 0{,}33$$

As you can see, symptoms must be classified as disease 1 if we use the *dice coefficient* as classifier. Also, care that *stiff neck* is a phrase. *Stiff* may be another symptom within the system. Then, if the symptom is '*stiff neck'*, do not make a mistake by dismissing '*neck'* keyword as if the symptom is just 'stiff'.

You are not allowed to use String methods of Java such as *contains, equals* etc. It will be penalised in the evaluation rigorously.

Secondly, your program will train the expected values for a casual day of each location. Assume that, first *n* days of log do not contain any outbreak for any infectious disease. That is, we pretend to know that, there is no outbreak for any disease in the first *n* days. *n* value will be given as described in the following section. Training will be performed statistically by the given steps:

- Parse log rows according to the location/disease/date of queries for the first *n* days
  - Queries contain location and date tags but you will classify the symptoms into diseases as described above.
- Calculate statistics of each location-disease couple; expected value (Eq.3) and standard deviation (Eq.4).

$$\varepsilon_{dx} = \frac{\sum_{j=0}^{n} y_{dj}}{n} \qquad \text{Eq.3}$$

$$sd_{dx} = \sqrt{y_{dxj}^2 - \varepsilon_{dx}^2} \qquad \text{Eq.4}$$

$d = disease\ d$

$x = location\ x$

ε_{dx} = expected case num. for disease d in location x

$sd_{dx} = \ std.dev.of\ case\ num.for\ diseased\ in\ loc.x$

$y_{dxj} = search\ count\ of\ disease\ d\ at\ time\ j$

$n = given\ window\ size\ to\ learn$

- Keep the expected values (mean) and standard deviations for the evaluation phase.

At the end of the learning step, you will have a list every element of which has to be modelled as follows (this is the minimum amount of information):

Lastly, your program will analyse the remaining of the log, from day *n+1* to the end of log file with Shewhart chart. If there is a shift greater than 2(Eq.5.1) within day t, location x for disease d, your program will write down the outbreak risk for x;t;d.

$$\frac{y_{dtx} - \varepsilon_{dx}}{sd_{dx}} \geq 2 \qquad \text{Eq.5.1}$$

$$y_{dtx} \geq \varepsilon_{dx} + 2sd_{dx} \qquad \text{Eq.5.2.}$$

where y_{dtx} is query count for the day *t*, location *x* and disease *d*. Care that; Eq.5.1. and Eq.5.2. are the same. Don't forget that, disease classification has to be done by your program from symptoms given into query as described above.

You will gain experiences on data mining subtitles such as *Text Mining, Statistical Process Controls, and Anomaly Detection.*

3. INPUT/OUTPUT

Two input files and an output file will be handled by your program. First input file is disease file that contains diseases and symptoms for each one. File format will be as given below:

<disease_name_0>:<symptom_00>,<symptom_01>,…,<symptom_0n_0)
<disease_name_1>:<symptom_10>,<symptom_11>,…,<symptom_1n_1)
⋮
<disease _name_m>:<symptom_m0>,<symptom_m1>,…,<symptom_mn$_m$)

*disease_name*s and *symptom*s can consist more than one word. Some symptoms can be shown while different diseases.

Sample disease file is:

typhoid:fever,bradycardia,malaise,headache,cough,dicrotic pulse
cholera:diarrhea,vomiting
flu:fever,extreme coldness,cold,cough,nasal congestion,runny nose,throat aches,joint aches,fatigue,headache,watering eyes
malaria:headache,fever,shivering,joint aches,vomiting,hemoglobin in the urine,retinal damage,convulsions
typhus:chills,fever,high fever,joint aches,rashes,headache,muscle pain,low blood pressure

Second input file is the health log file provided by online search engine. Each row of this log consists of; timestamp of the search; location information that could be a city name such as Ankara, Sivas etc.; and lastly keywords of the search text extracted from raw search text by the search engine. Each row ends with End of Query character (\Q).

<timestamp><location>$<keyword_0>$<keyword_1>$…$<keyword_n><EndOfQuery>

Sample query log file:

...28/05/2013Amasya$chills$joint$aches$runny$rose$stiff$neck$extreme$coldness\Q28/05/2013Konya$asthenia\Q28/05/2013Istanbul$watering$eyes$asthenia$throat$aches$dicrotic$pulse$retinal$damage\Q28/05/2013Mardin$headache$asthenia$hemoglobin$in$the$urine$muscle$pain\Q28/05/2013Adana$shivering\Q28/05/2013Rize$joint$aches\Q28/05/2013Sivas$vomitting$dicrotic$pulse$dierrhea$asthenia$low$blood$pressure\Q28/05/2013Rize$muscle$pain$malaise$fatigue$asthenia$chills\Q28/05/2013Mardin$extreme$coldness$malaise\Q28/05/2013Adana$vomitting\Q28/05/2013Istanbul$rashes$watering$eyes$stiff$neck\Q28/05/2013Amasya$dierrhea$joint$aches\Q29/05/2013Istanbul$fever$throat$aches$nasal$congestion\Q29/05/2013Sivas$muscle$pain\Q29/05/2013Adana$runny$rose$headache$fever$rashes$convulsions\Q29/05/2013Izmir$joint$aches$retinal$damage$shivering\Q29/05/2013Konya$runny$rose$fatigue$cold\Q29/05/2013Amasya$shivering\Q29/05/2013Konya$asthenia$shivering$retinal$damage$rashes\Q29/05/2013Istanbul$fever\Q...

Colouring is being done to make sample log file section more understandable. As you can see in the sample log, */Q* special character defines the end of a query row. Also timestamp format is dd/mm/yyyy. *$* Character severs the keywords. Be careful that, *$* character is placed at the beginning of the keyword, not at the end of it. Your application will run from the command line with such a call given below:

Java main.java <disease file> <log file> <n value> <output file>

Your program will create diseases-symptoms associations from disease file. After that, your program will train with queries belongs to first n day. Assume that, *n* value is 10 and your program labelled symptoms as flu for Ankara during 10 days respectively: 15,17,20,14,22,12,14,16,19,16. Your program should calculate expected value and standard deviation of flu trends in Ankara as:

Expected value is the average of the labelled count of query from Eq.3.:
(15+17+20+14+22+12+14+16+19+16)/10 = 16.5
Standard deviation computed from Eq.4. is: *3.06*

You will have a testing model built with training dataset extracted from query data set. At last step, your program will test remain of the queries with the statistical model built previous phase. Assume that, your program has statistics provided previous step as given above and there is 25 query rows labelled as flu for Ankara at day *29.05.2013*. Your program will test the statistics:

is **25** greater than statistic computed with Eq.5.2 ?
16,5+2x3,06 = **22,62**

The test statistic says, there is a flu outbreak in Ankara at day *29.05.2013*. Your program should write down the outbreak to output file as:

flu detected in Ankara at 29.05.2013:25-22,62

Template of the output format is:
<disease> detected in <location> at <timestamp> <observed value> <threshold>

Threshold is right-hand side of the Eq.3.2. It is sum of *expected value* and two times of *standard deviation* ($\varepsilon_{dx} + 2sd_{dx}$).

4. SUBMISSION

<student_id>
    <report>
        report.pdf
    <src>
        main.java
        *.java

5. NOTES AND RESTRICTIONS

- You will use online submission system to submit your experiment. **No other submission method** such as CD, USB, e-mail will be accepted.

- Submission time for deadline is 17:00. Submit system will stay open until 23.59 at deadline**, but any problem you faced after 17:00 will be under your responsibility**. We have no physical access chance during the evening.
- Do not submit any file via e-mail related with this assignment
- Save and don't share all your work until the assignment is graded announced at the end of reclamation period.
- The assignment must be original, **INDIVIDUAL** work. **DUPLICATE** or **VERY SIMILAR ASSIGNMENTS** are both going to be punished rigidly. General discussion of the problem is allowed, but **DO NOT SHARE YOUR DESIGN OR IMPLEMENTATION**.
- You can ask your questions through course's piazza page:
    - https://piazza.com/hacettepe.edu.tr/spring2013/bbm204/
- You are supposed to be aware of everything discussed in this page

6. REFERENCES

[1] «The School of Information Studies,» [Online]. Available: http://infospace.ischool.syr.edu/2013/04/23/what-is-text-mining/.

[2] «Wikipedia,» [Online]. Available: http://en.wikipedia.org/wiki/Anomaly_detection#cite_note-1.

[3] «Wikipedia,» [Online]. Available: http://en.wikipedia.org/wiki/Outbreak.

[4] W. A. Shewhart, « Economic Control of Quality of Manufactured Product,» 1930.

[5] «Google Grip Trendleri,» [Online]. Available: http://www.google.org/flutrends/about/how.html.

[6] M. A. Hearst, «Text data mining: Issues, techniques and the relationship to information access.Presentation notes for UW/MS workshop on data mining,» 1997.