UNIVERSITY OF AGDER

Spam classification for online discussions

By

Hao Wu

Thesis submitted in Partial Fulfillment of the
Requirements for the Degree Master of Technology in
Information and Communication Technology

Faculty of Engineering and Science
University of Agder

Grimstad
May 2010

Abstract

Traditionally, spam messages filtering systems are built by integrating content-based analysis technologies which are developed from the experiences of dealing with E-mail spam. Recently, the new style of information appears in the Internet, Social Media platform, which also expands the space for Internet abusers.

In this thesis, we not only evaluated the traditional content-based approaches to classify spam messages, we also investigated the possibility of integrating context-based technology with content-based approaches to classify spam messages. We built spam classifiers using Novelty detection approach combining with Naïve Bayes, k Nearest-Neighbour and Self-organizing map respectively and tested each of them with vast amount of experiment data. And we also took a further step from the previous researches by integrating Self-organizing map with Naive Bayes to carry out the spam classification.

The results of this thesis show that combining context-based approaches with content-based spam classifier wisely can actually improve the performance of content-based spam classifier in variant of directions. In addition, the results from Self-organizing map classifier with Naïve Bayes show a promising future for data clustering method using in spam filtering.

Thus we believe this thesis presents a new insight in Natural Language Processing and the methods and techniques proposed in this thesis provide researchers in spam filtering field a good tool to analyze context-based spam messages.

Preface

Thesis submitted in Partial Fulfillment of the Requirements for the Degree Master of Technology in Information and Communication Technology at University of Agder, Norway. The project is supported by Integrasco A/S, who has provided data material and supporting frameworks which have been used to carry out various parts of this study. This work was processed under the supervision of Jaran Nilsen at Integrasco A/S, Norway and co-supervisor associate professor Ole-Christoffer Granmo at the University of Agder, Norway.

First of all, I would like to thank my supervisor Jaran Nilsen (Integrasco A/S), who uses his experience and expertise in the field of Computer Science to help me throughout project period. He's also a friend with patience who inspires me to believe in myself. Without his help, I would not have finished in time. And I wish to thank professor Ole-Christoffer Granmo for his assistance and support through the project period. His professional insight on Natural Language Processing field helped me a lot on understanding background theories. And here I also wish to thank my families who support my study in Norway and have given countless encouragements from thousands of miles away in China. Last but not least I would like to thank my fellow student and friend Stian Berg who provided lots of valuable advices on my prototype design and writing.

Grimstad, May 2010.

Hao Wu

Contents

Chapter 4

Chapter 5

Chapter 6

List of Figures

List of Tables

Chapter 1
Introduction

Back to the old times, when we have watched a drama or a sport game we would prefer to talk to our neighbours and friends as soon as possible. We believe that we could share our happiness with others in the talks and discussions. When we need some helps such as clogged pips or babysitting, those neighbours would be our first choice before other options run through the head. However, those have all been changed by a great invention “computer”.

None of us is able to escape from this life style revolution caused by the birth of computer and everyone is involved in this significant change due to the Internet. In this project, we will show you how significant it is and what can we do to adapt to it.

1.1 Problem declaration

Have you being wondering where are those sales men going? They used to knock our doors and start with a very polite asking “Excuse me, can I have you a few seconds?” Those “a few seconds” will soon be increased to a couple of minutes. If you don’t stop them and throw them out of your house immediately, you will be out of your mind by their endless stories. Even if they don’t visit you in the front door, they could also “concern” your needs by calling you. This situation is even worse that it is unpreventable before you pick up the phone. However, why don’t we have sales calls and door-to-door sales men anymore? Are they still in the same time and space where we stay? Yes, they do. They just changed the way to make business. We shall solve the mystery in the following parts.

Living in the 21th century, we no longer enjoy those great afternoons with friends and families by drinking a cup of coffee or tees. After a tough day, we are more likely to join in those conversations and discussions what we concern through Internet. There we have a lot of virtual communities which promise to provide nice and safe places for us to give opinions.

Without face-to-face conversation, participants may be in the different age groups, social classes or different places. They have chances to talk to whom they hardly have chance to meet during the work or in their normal life or even in their spaces. In this case, the platform gives every participant an equal opportunity to present ideas and thoughts. It also gives opportunities to someone who doesn’t intent to participate in discussions.

As the number of Cyber Citizens is growing rapidly in these recent years, Internet brought us much more surprises than we used to expect, not only explosive information but also explosive problems. On one hand, people enjoy the good gossip with some John Doe from any part of the world; on the other hand, sales men are trying their best to interrupt into your conversations to get as much attention as possible.

Now, we may give this kind of sales men a new name “spammer”. Once we all clear about the purpose of spamming and how they will spam, we could start to think about the solution to filter spam from those platforms.

Consider these two situations: 1. sales men visit every house in the community taking a couple of weeks; 2. sales men send messages which contain every detail of the product to a discussion which will be browsed by dozens of people in a minute. Which one is better in your opinion? In our opinion, there is no doubt that the second one is way better than the first one which not only saves the time and energy for the sales men but also gives guarantee to products from being damaged during the demonstration travelling.

From the above discussion we could understand that why sales men pick Internet to broadcast their advertisements. There is no fault to take a channel as low-cost as possible to do business in the first place. However, it is unacceptable that if the business has interrupted into our normal life. Those sales men not only broadcast their advertisements they actually push people to read them and this is the reason we call them — spam. Then we shall know how these spammers spam our Internet.

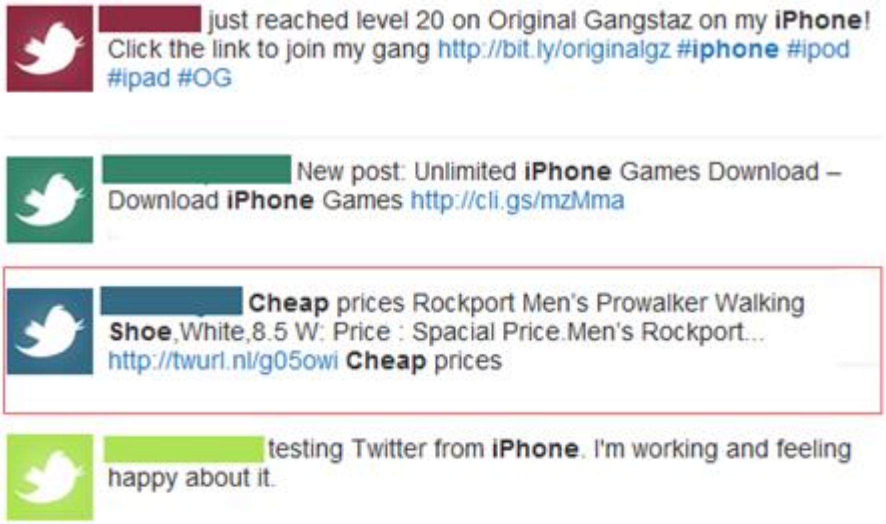
There are several places that we can find spam in the Internet such as E-mails, news groups, forums and etc. If you want to broadcast advertisements to as many people as possible, you need to choose the most crowd places and Social Media Conversation (SMC[1]) is one of those places.

In this project, SMC as a main channel of spamming would be our main research channel as well. Where there are SMCs, where we can develop our research. Forums, blogs, tweets[2] are all parts of SMCs. In this project, we may use forum discussions as we our primary resources. Here we present some situations which we may encounter when we browse do SMCs.

First one, also the most familiar one to us, is that someone posts totally different content to the subject, the post which perhaps contains an advertisement of some products or stories to tempt donations.

[1] We will use SMC instead of Social Media Conversation in the following pages.

[2] Tweets is a new type of SMC.



[redacted] just reached level 20 on Original Gangstaz on my **iPhone**! Click the link to join my gang http://bit.ly/originalgz **#iphone** #ipod #ipad #OG

[redacted] New post: Unlimited **iPhone** Games Download – Download **iPhone** Games http://cli.gs/mzMma

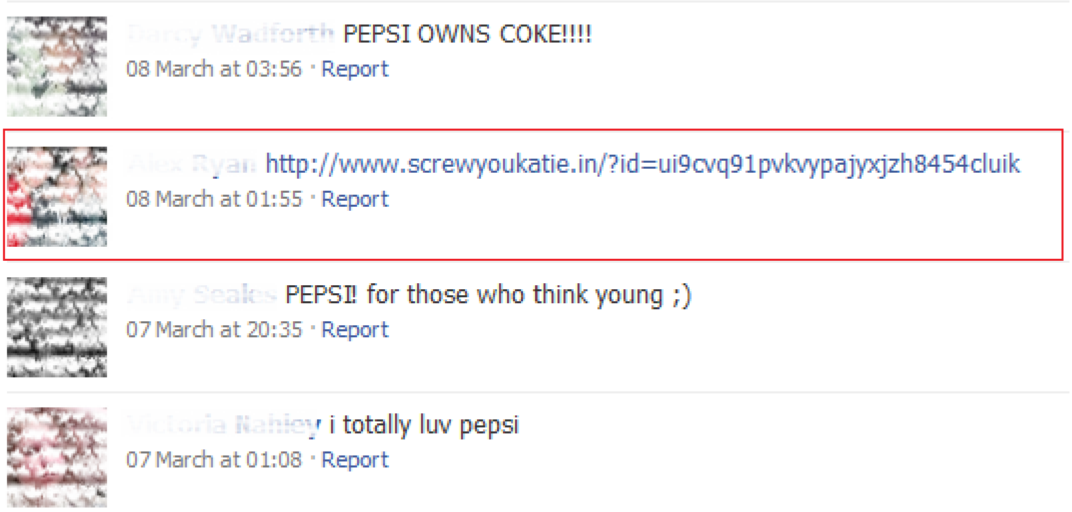
[redacted] **Cheap** prices Rockport Men's Prowalker Walking **Shoe**,White,8.5 W: Price : Spacial Price.Men's Rockport... http://twurl.nl/g05owi **Cheap** prices

[redacted] testing Twitter from **iPhone**. I'm working and feeling happy about it.

Figure 1-1 SMC spam example 1

For instance, Figure 1-1 shows a discussion which took place in some twitter[3] user's wall. They were apparently into the topic of iPhone. Due to the limitation on space, we couldn't afford to quote every post here. So, we only pick out a section of this discussion which contains four replies. As the red rectangle indicating, this replier wrote something, about cheap shoes, which was totally irrelevant to the whole topic. In this case, we could consider the reply is a spam.

The following one, also familiar to us, is that someone posts external links in the reply which attracts people to click by saying nothing or saying it's helpful to the subject, however, it is not always the case.



[redacted] PEPSI OWNS COKE!!!!
08 March at 03:56 · Report

[redacted] http://www.screwyoukatie.in/?id=ui9cvq91pvkvypajyxjzh8454cluik
08 March at 01:55 · Report

[redacted] PEPSI!! for those who think young ;)
07 March at 20:35 · Report

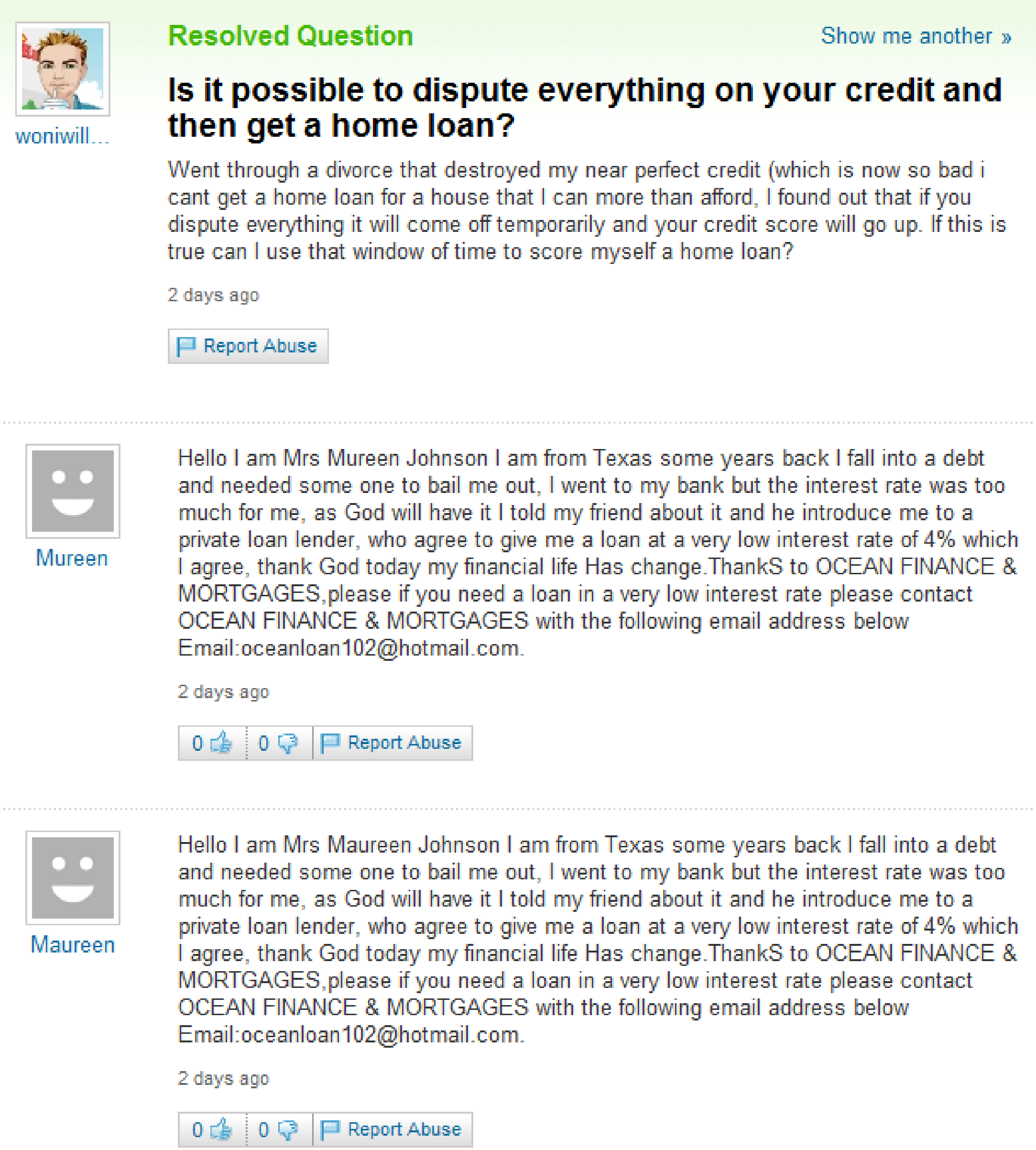
[redacted] i totally luv pepsi
07 March at 01:08 · Report

Figure 1-2 SMC spam example 2

[3] Twitter is the place where people post opinions or tweets each other.

Here we show an example in the above Figure 1-2, with the “help” of a spammer, in which we may have a clear impression about in what way and where a spammer will post messages. The discussion is a fragment of large discussion happened on the wall page of a Facebook[4] group user. All those posts are about the topic referring to “Pepsi” excluding the second reply which had only a link address. Apparently, the replier wanted participants to believe that the web site is something related to the topic. However, we found out that this web site had nothing to do with Pepsi after we tested it. Therefore, we consider this post is a spam.

The last one is the most challenging situation which we will encounter in this project. In this situation, the spammer post messages which are relevant to the topic started by the first person, but the purpose of the posting is to publicizing a certain service or product of their own.



woniwill...

Is it possible to dispute everything on your credit and then get a home loan?

Went through a divorce that destroyed my near perfect credit (which is now so bad i cant get a home loan for a house that I can more than afford, I found out that if you dispute everything it will come off temporarily and your credit score will go up. If this is true can I use that window of time to score myself a home loan?

2 days ago

Mureen

Hello I am Mrs Mureen Johnson I am from Texas some years back I fall into a debt and needed some one to bail me out, I went to my bank but the interest rate was too much for me, as God will have it I told my friend about it and he introduce me to a private loan lender, who agree to give me a loan at a very low interest rate of 4% which I agree, thank God today my financial life Has change.ThankS to OCEAN FINANCE & MORTGAGES,please if you need a loan in a very low interest rate please contact OCEAN FINANCE & MORTGAGES with the following email address below Email:oceanloan102@hotmail.com.

2 days ago

Maureen

Hello I am Mrs Maureen Johnson I am from Texas some years back I fall into a debt and needed some one to bail me out, I went to my bank but the interest rate was too much for me, as God will have it I told my friend about it and he introduce me to a private loan lender, who agree to give me a loan at a very low interest rate of 4% which I agree, thank God today my financial life Has change.ThankS to OCEAN FINANCE & MORTGAGES,please if you need a loan in a very low interest rate please contact OCEAN FINANCE & MORTGAGES with the following email address below Email:oceanloan102@hotmail.com.

2 days ago

Figure 1-3 SMC spam example 3

[4] Facebook is a virtual society where people share information, etc.

You may be confused that why we duplicated the replies for this discussion in the above graph. The truth is we didn't. Yes, they are almost the same if you don't pay 100% attention to them, but soon you will notice that names of these posters are different, "Mureen" and "Maureen", just one letter difference. In this given example, the questioner encountered a problem with the credit card. The following responses seem quite rational which tell a story of her own with the similar situation and offer a contact to the poster that can solve his problem. If there is only one response with this attractive story, the questioner may have no doubt to try that contact, but those replies have been deleted as we speak. The suspicious part of these replies is that they have the same content if we ignore the signature which makes them untrustworthy. According to the experiences of dealing with E-mail spam, we can affirm and have no doubt that these replies are spam.

These above three situations are considered as Forum spam. Our intention is to find out an algorithm to automatically and efficiently filter the spam posts from discussion board.

Now, what we have are algorithms which work well on E-mail spam classification. There are several common places easily shown between E-mail spam and Forum spam. In the first place, both E-mails and SMCs are written by nature languages. Secondly, their target users are both Internet users. And thirdly, spammers in both circumstances may have some common places such as they use the convenience of Internet to broadcast their thoughts.

1.2 Challenges

Based on the above observations, we attempt to transplant those algorithms, which work well on E-mail, from E-mail spam classification to Forum spam classification. But there is no free way to go that we have to face several challenges during the project.

One of the challenges of this thesis is that there are still huge differences between E-mails and Forum posts:

- Forum posts are usually much shorter than normal E-mails which means posts usually contain much less words and information than E-mails
- Forum posts are not always informative or meaningful when they are treated individually, i.e. they are meaningful when they stay in the thread context

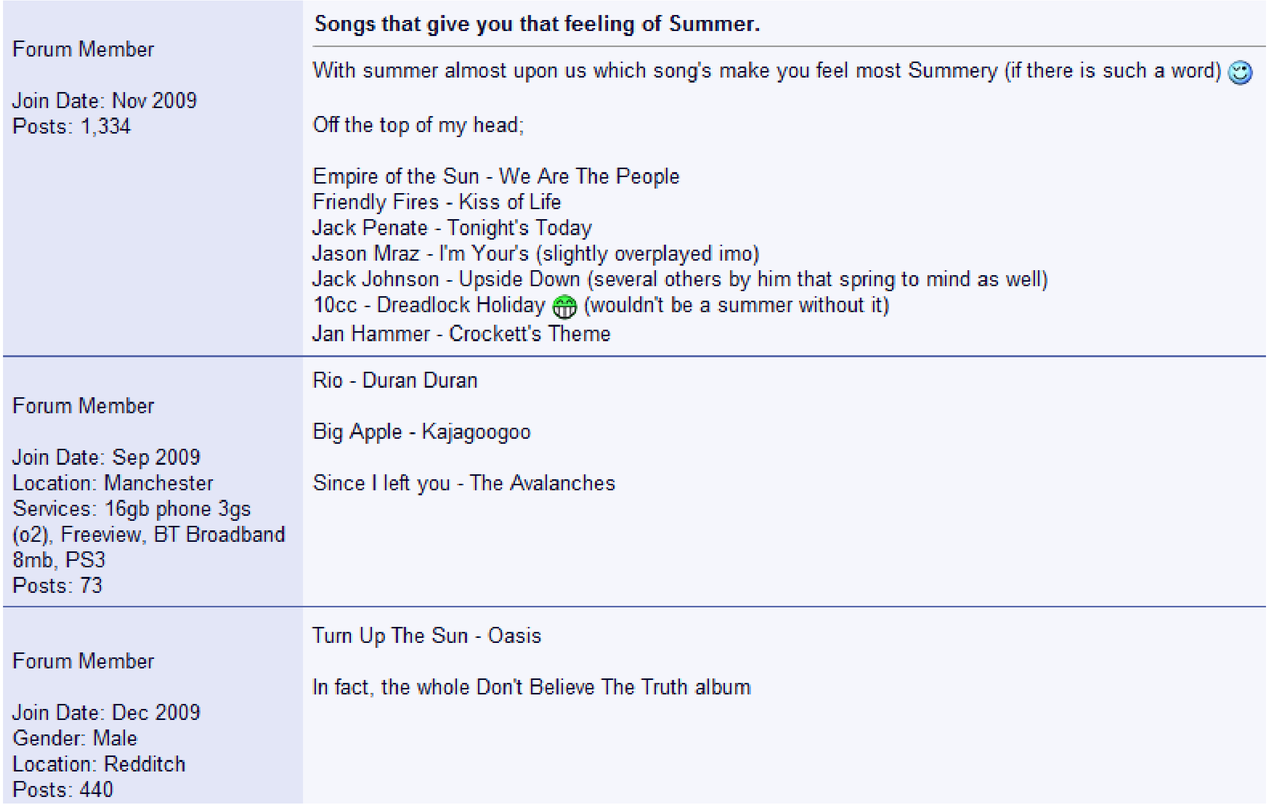
To illustrate these two differences between E-mails and online discussions, we introduce the following example in Figure1-4 which is found in one of the UK forums and the topic from the initiator is "Songs that give you that feeling of Summer.".

Since the title of this thread is about songs, so we could easily find out that the two replies are both associated to songs. Therefore, according to the reasons we presented in above sections, we consider that both of these two replies are ham posts.

Normally the discussions in online discussion board are short and reply to the topic or other repliers just like the discussions in this example. Comparing this to ordinary E-mails where people could hardly express a concrete idea or complete story in such a few words, we believe that we have demonstrated clearly in the difference of E-mails and forum posts regarding to the length of writing.

Then we also could illustrate the second difference which is regarding to the informative issue of writing with this same example. From this thread in Figure 1-4, we find that the second and third replies both have only written something replying to the topic but without mentioning or referring to the topic in the writing.

If we read the last two replies separately, we may get nothing but some names of songs and singers and we will not know that they are discussing about songs which have summer taste. Or, maybe we will even consider they are both some kind of advertisements if they appear in other threads which are not related to music. So back to this thread, we understand the reason that why we say “forum posts are not always informative or meaningful when they are treated individually”.



Forum Member Join Date: Nov 2009 Posts: 1,334	**Songs that give you that feeling of Summer.** With summer almost upon us which song's make you feel most Summery (if there is such a word) ☺ Off the top of my head; Empire of the Sun - We Are The People Friendly Fires - Kiss of Life Jack Penate - Tonight's Today Jason Mraz - I'm Your's (slightly overplayed imo) Jack Johnson - Upside Down (several others by him that spring to mind as well) 10cc - Dreadlock Holiday 😁 (wouldn't be a summer without it) Jan Hammer - Crockett's Theme
Forum Member Join Date: Sep 2009 Location: Manchester Services: 16gb phone 3gs (o2), Freeview, BT Broadband 8mb, PS3 Posts: 73	Rio - Duran Duran Big Apple - Kajagoogoo Since I left you - The Avalanches
Forum Member Join Date: Dec 2009 Gender: Male Location: Redditch Posts: 440	Turn Up The Sun - Oasis In fact, the whole Don't Believe The Truth album

Figure 1-4 Sample online discussion

Apart from the challenge brought by the differences between E-mails and online discussions described above, we also have other challenges such as the challenge brought by ordinary online discussions themselves.

As we presented in the above sections, some spam look quite similar to ordinary posts, such as the one in Figure 1-3. So, classifier may mistakenly classify ordinary posts into spam.

Since our task is to classify online discussion posts or to say is to classify spam posts from ordinary posts, people may place ordinary posts in the second place. However, even though we are focusing on spam, the importance of ordinary posts cannot be ignored. In the other direction to describe this issue is that we couldn't lose information or data by carrying out the spam classification.

Under the above illustration, we may show the challenges very clearly that we are facing in this thesis. And then, by considering the situations and aspects will be encountered in this project, we will try to implement machine learning algorithms into forum spam classifiers and make modifications properly on the usages of algorithms to make them suitable for forum posts.

1.3 State of art

In the past few decades, researchers have already put lots of strength on E-mail abuse and successfully developed kinds of methods to prevent the E-mail spam distribution. For instance, researchers in [1] proposed a method using boosting tree to filter spam Email where AdaBoost gives quite good result for spam filtering. And researchers in [2] compared Naïve Bayes classification to Memory-based classification for spam filtering with the conclusion that both approaches gain very high classification accuracy and memory-based approach appears to be more viable by careful configuration.

Recently, as the Internet stepping into hundreds of thousands of ordinary people's life, the abuse of social media to advertise or spread bulk information attracts researchers' attention. Researchers from UMBC are doing great work in Social Media spam investigations as they presented in [3], the spam messages are growing all the time through all new kinds of Social Media channels. In 2007, scientists have already tried to filter spam messages from forums in [4] in which presented us a quite good context-based approach to detect spam.

That even some researchers have already started to analyze Social Media spam, it's still a quite new topic and field to be explored.

1.4 Motivation and Definition

What will motivate us to do the research in this project? We will start by talking about the significance of spam classification for SMC.

Our master thesis is offered by Integrasco A/S. Integrasco A/S is mining vast amount of data from online discussion boards. From time to time these discussion boards consists of several spam "infected" discussions. Being able to filter out this spam would help increase both the readability of the discussions, limit the storage requirements for the mined data, and also help prevent the spam from confusing automatic analysis algorithms trying to understand what is being discussed.

As a network user, we need a clean area to discuss our interests without disturbance; as a trend analyst, we need our working space without spam which will highly increase our efficiency to analyze data; as the product given to our customers, we need to make our mined data more readable and understandable. By the accomplishment of this project, it will have something to do with the improvements of Integrasco A/S data retrieval system if it has been applied to the product.

Then we give the following definition to this project:

The purpose of this study is to find an efficient approach to classify spam posts from conversations in social media. Discussions from online discussion boards will be used to test our theories. As a result, spam posts will be filtered from ham[5] posts automatically by applying different machine learning and pattern classification techniques.

In the project, work will focus on three different machine learning algorithms which are k-NN (k Nearest-Neighbour), Naive Bayes and SOM (Self-organizing Maps). After the investigation of each of these algorithms, we will first try to implement content-based approach and to analyze which algorithm is more suitable than others in spam message classification in discussion boards. Then, we will try to implement context-based approach with each of these algorithms to see if there any improvements can be made by combining with the approach.

Spam E-mails which can be considered as a context-independent conversation usually tell a complete story itself. We only need to consider the E-mail itself when we intend to classify it into spam and ham category. When it comes to SMCs, unlike E-mail, SMC always consist of many dialogues. A single comment may look like a spam without the context in the conversation, however, which can be really relevant to the topic. For instance, a comment which says “check out this super sale for N97 http://xxx.xxx.xxx” may be considered as a spam by itself from the above description, but if the conversation is talking about the cheapest phone in the town then the comment changes to be a much valuable one. In this case, here the challenge for us to do the spam classification for SMCs that is we need to put the whole conversation under consideration not only a single comment.

For evaluation purpose, spam posts and ham posts from online discussions obtained from Integrasco A/S will be used. At the end, accuracy of each three algorithms will be employed to evaluate them in spam classification and pattern recognition field for social media, especially for online discussions.

1.5 Academic contribution

In this thesis, we evaluate three machine learning theories Naïve Bayes, k Nearest-Neighbour and Self-organizing map in spam message classification for SMC. In the comparison among these three machine learning theories, we employ strict laboratory experiments on vast amount of data imported from real world to demonstrate the process and results of spam classification.

[5] Ham is used to describe non-spam posts.

To the best of our knowledge, previous researches in spam message detection and filtering mostly focus on content-based environment or to say context-free environment which means in the classification, classifier always treats each spam message as an individual object among other experiment data. We also evaluate the efficiency based on context-free environment spam message classification from the start of our work, but more importantly we introduce a context-based approach to carry out the classification. According to the results from our experiments, we have confidence to believe our approach could highly improve content-based spam message classification in SMC.

This thesis also contributes prototypes which implement content-based spam message classification and context-based spam message classification which will be interested to researchers who dedicate in spam filtering and detection field.

1.6 Target audience

The target audience of this report is anyone interested in users' behaviours and opinions in Social Media Conversations or spam classification. The report is written in a language that requires fundamental knowledge of probability and machine learning theories with standard computer programming implementation so experience of computer programming is advised. However, the analysis and results coming along presented in this report require no insight into computer programming or related theoretical background. The accomplishment in this report is also related to web data analysis and nature language processing especially in online discussion boards therefore this report is relevant to someone working in the corresponding fields.

1.7 Report outline

The rest of this report is organized as follows:

Chapter 2 Explains the background theories used in this thesis

Chapter 3 Presents the details of algorithms used to develop prototypes

Chapter 4 Gives a brief view of prototypes as well as some results

Chapter 5 Discusses the performances of classifiers

Chapter 6 Draws a conclusion of this thesis work

Chapter 2

Pattern recognition & Novelty detection

2.1 Pattern Recognition

How can we possibly recognize that these animals (left four) in the following Figure 2-1 are dogs whereas they ware different kinds of fur respectively and each of them may bark in an unfamiliar tune?

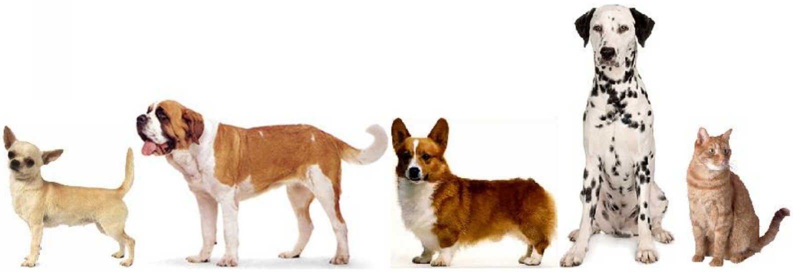


Figure 2-1 Example of pattern recognition

As human being, according to the long term of observations of these animals, we have an empirical conclusion passed down generation by generation which is they all have sharp nose, long ear, tongue panted and tail up. These brief descriptions may not include all features of dogs but at least you can distinguish dogs from cats (the most right one in the above Figure 2-1) by above observation results.

To summarize the technique used to answer the above questions, we introduce a term: Pattern. Pattern is the feature abstracted from a cluster of similar objects which repeatedly show up in a predictable way. On the contrary, if there are lots of objects which have the same pattern then we could form a group for them. [5]

2.1.1 Importance of Pattern recognition

Sometimes we need to arrange lots of things, such as we need to recognize dogs and cats from images. Human beings like us will lean on the empirical conclusion which has already been described above to accomplish the job. However, if there are a mass of images of dogs and cats which we are not able to check them one after another, the task to distinguish them will touch the ceiling of our capacity and may take a long time to finish. What if these tasks are needed to be done fast? We need to find a helper who can do it much faster than us and without mistakes.

Assuming we want to locate stray dogs and cats in the city. By utilizing those images captured from surveillance cameras, we could easily obtain vast amount of objects moving around. To simplify the case, we assume the objects are only dogs and cats.

Even though dogs and cats can be easily recognized from images by human beings, they are just pixels to computer or to say it even more specifically they are just zeros and ones. How could a computer recognize the whole object by looking at one pixel a time? Without the overview, a pixel can represent everything.

Only if pixels or groups of pixels can be used as symbolic features indicating either dog or cat, the computer could process classification work. To teach computer to learn something on its own is also the most challenging part of machine learning that means to teach computer to learn those features or even discover features by themselves.

2.1.2 Pattern recognition and spam classification

If dogs and cats are still countable to us, SMCs from Internet will be “uncountable” in this information explosive period. Can we filter out those spam messages by our own hands? Simply to say in the thread within a forum, spam classification is to filter those conversations described in the previous chapter from normal conversations which are part of the thread. Maybe we can, but in my opinion it will take vast amount of time which we cannot afford.

Empirically also from the natural of spamming, to save time and energy, spammers will not create lots of totally different content for a certain aim. For instance, if they want to tell a story about credit card just like the example given above, they will use a template to broadcast their information only with different signatures. This feature leads us to think that there will be a fixed pattern for each kind of spamming or spam messages.

After we find those patterns and make them understandable procedures for machines, we will teach the computer to follow those procedures to carry out the classification. The products in the spam message may be changed, the contact in the spam message may be also changes, but the goal of spammers, to make money from nothing, will not change.

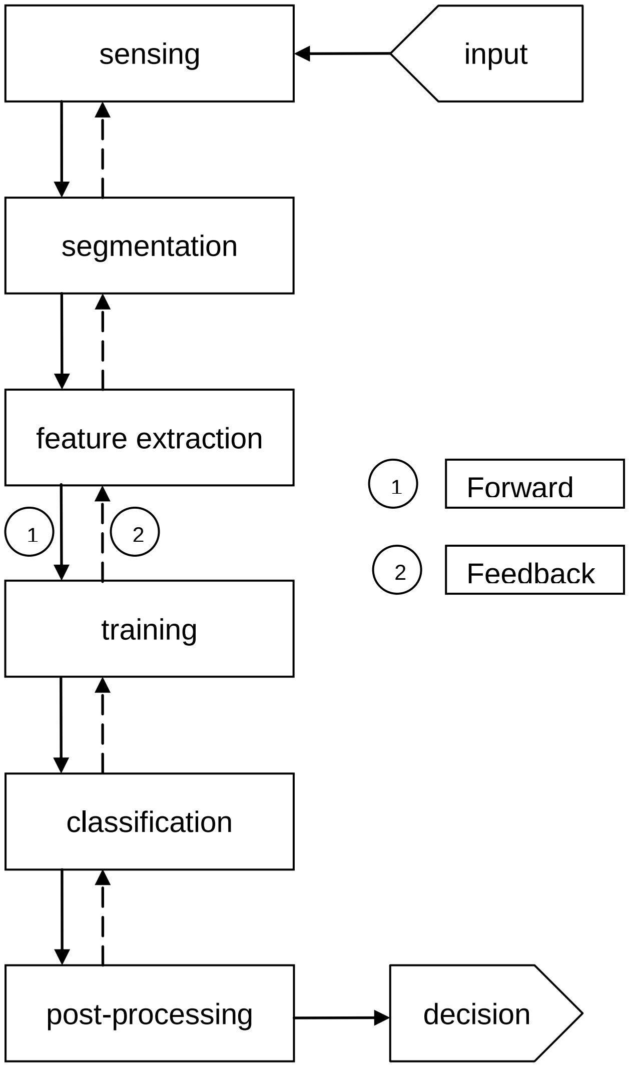
2.1.3 Pattern classification implementation

“People are often prone to making mistakes during analyses or, possibly, when trying to establish relationships between multiple features.” [6]

We have too much optical and acoustical information going through our input system to make us hardly remember some simple figures. However, with the help of machine, we could store as much information as we designed.

The flowchart below shows a simplified process for pattern classification. From the top right to the bottom right, the system or classifier starts from feed in raw data till the end of categorization of each individual object.

In our project, thousands of forum posts will be fed into the classifier along with feature extraction process using words or tf-idf value as features. Training phase would be necessary when we use supervised learning.



Pattern recognition is a part of machine learning which plays an extremely important role all around us such as vending machines or broker software.

All of the objects in a given dataset used by machine learning algorithms are represented using the same vector of features. These features may be continuous, categorical or binary. [Supervised Machine Learning: A Review of Classification Techniques]

If the object is given with known labels as well as others in the same dataset, the corresponding learning is called supervised learning.

Opposite of supervised learning, if the object in the dataset has no label with it, the learning process will be called unsupervised learning.

In the middle of supervised and unsupervised learning, there is another machine learning method which utilizes the dataset with some labelled data and some unlabeled data call semi-supervised learning.

A third machine learning also known as reinforcement learning gets information from external providers. The external providers will only applaud or criticize how the learning goes along.

2.2 Novelty Detection

To a document retrieval system, the main task is to find documents related to user's information need. And there is a challenge to retrieval system that is to show documents not only related to user's need but also including something new which has never been seen by user.

Novelty track 2002 [7] expressing the situation described before has been a long term problem for information retrieval systems [8]. From 2002, novelty detection has been put on the table and researchers harvest a lot since that time.

The intension to improve pattern classification algorithms drives us to combine novelty detection method with those algorithms associate with TF-IDF and cosine similarity described next. In this project, the novelty detection method will serve as in the post-process phase after classification phase.

2.2.1 TF-IDF

TF-IDF (term frequency-inverse document frequency) is a weight used to measure the value of a document in the document set which works as the feature extractor in our project.

Term frequency is used to measure how valuable a term in the document. In the following formula 2.1, the higher the frequency of the term the higher value will be assigned to this term.

$$\mathrm{tf}_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}} \tag{2.1}$$

Where $n_{i,j}$ indicates the number of occurrences of term (t_i) in document (d_j) and the denominator is the total count of occurrences of all terms in document (d_j).

Inverse document frequency is a measurement to represent the frequency of a document containing a certain term. By the beneath calculation, the document which contains new terms will be highlighted, on the contrary, documents have most similar content will be assigned to a low weight.

$$\mathrm{idf}_i = \log\frac{|D|}{|\{d:t_i \in d\}|} \quad (2.2)$$

Where $|D|$ is the total count of documents in the document set and the denominator is the number of occurrences of documents containing term (t_i).

$$(\text{tf-idf})_{i,j} = \mathrm{tf}_{i,j} \cdot \mathrm{idf}_i \quad (2.3)$$

In this way, a distinctive document will earn a high tf-idf value within all documents in the document set which can be used to detect the novelty of a retrieved document. In addition, this value is usually normalized in [9] when documents are in variant length.

$$w_{i.j} = \frac{\mathrm{tfidf}\,(t_i, d_j)}{\sqrt{\sum_{s=1}^{|T|}(\mathrm{tfidf}\,(t_i, d_j))^2}} \quad (2.4)$$

Where $w_{i.j}$ is the weight assigned to document (d_j) and the denominator is a square root of the sum of tf-idf values of all terms from term (t_1) to term $(t_{|T|})$ in document (d_j).

2.2.2 Euclidean distance

Euclidean distance is widely used by researches when the distance between two vectors is needed to be calculated. Now, if exists vectors $A = [a_1, a_2, a_3, \dots, a_m]$ and $B = [b_1, b_2, b_3, \dots, b_n]$ when $m = n$ we have:

$$\mathrm{dis}_{\mathrm{Euclidean}} = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + (a_3 - b_3)^2 + \cdots + (a_m - b_n)^2} = \sqrt{\sum_{i=1}^{m}(a_i - b_i)^2} \quad (2.5)$$

2.2.3 Cosine similarity

In the vector-space area, we use Euclidean distance to measure the difference between two points. After the calculation of tf-idf value for each document, we also need to know the Euclidean distances among documents in order to find out the most distinctive ones or to say the similarity among them.

Cosine distance [10], so called cosine similarity, will be employed to compute similarity. We use tf-idf value as the weight of each document.

$$\mathrm{Cos}(d_i, d_j) = \frac{d_i \cdot d_j}{\|d_i\| \cdot \|d_j\|} \quad (2.6)$$

Where $\mathrm{d_i}$ or $\mathrm{d_j}$ are normalized tf-idf values for document $(\mathrm{d_i})$ or $(\mathrm{d_j})$ and $(\|\mathrm{d_i}\|)$ or $(\|\mathrm{d_j}\|)$ are magnitude tf-idf values for document $(\mathrm{d_i})$ or $(\mathrm{d_j})$.

2.2.4 Linear regression

Linear regression line is employed in this project to help us to measure the similarity between each post and the subject of the thread. We assume each post in the thread discusses around the same or similar subject, so the spam post which is not related to this subject will be placed far away from the regression line of this thread.

Given a set of points $\{(x_1, y_1), (x_2, y_2), (x_3, y_3), \dots, (x_n, y_n)\}$, the goal is to find a straight line Eq. (2.7) which goes through these points to make the sum of squared residuals of the model as small as possible.

$$y = \alpha + \beta x \tag{2.7}$$

We could make use of calculus to compute the parameters of this equation:

$$\hat{\beta} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = r_{xy}\frac{s_y}{s_x} \tag{2.8}$$

$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x} \tag{2.9}$$

where r_{xy} is the sample correlation coefficient between x and y, s_x is the standard deviation of x and s_y is the standard deviation of y, and $\bar{x}$ and $\bar{y}$ denote the mean values of x and y.

Chapter 3 Algorithms

3.1 Naïve Bayes

The Naive Bayes classifier which is based on applying Bayesian Theory could be trained very efficiently in a supervised learning setting.

Specifically, the Naive Bayes classifier assumes that the existence (or non-existence) of a certain characteristic of a class has no influence to/from the existence (of non-existence) of any other characteristic.

Considering the example we presented above, an object may be known as a dog if it has sharp nose, long ear, tongue panted and tail up. Even if these characteristics are all related to dogs and depended on presence of other characteristics, a Naïve Bayes classifier considers all these features to independently contribute to the probability that this object is a dog.

3.1.1 Bayesian theory

The example is just a brief introduction for the classifier, and then we need to explain the details. Assuming the object belonging to a certain class has several features, and then we use $\mathrm{F_i}(1 < i < n)$ to denote the feature and use C to denote a certain class.

Following the Bayesian Theorem, we have the expression below:

$$\mathrm{p(C|F_1, \dots, F_n) = \frac{p(C)\, p(F_1,\dots,F_n|C)}{p(F_1,\dots,F_n)}} \tag{3.1}$$

Since these features are dependent between each other and the denominator is decided by C, so we could rewrite the numerator like this:

$$\begin{aligned} &\mathrm{p(C, F_1, \dots, F_n)} \\ &= \mathrm{p(C)\; p(F_1|C)\; p(F_2|C, F_1) \cdots p(F_n|C, F_1, F_2, \dots, F_{n-1})} \end{aligned} \tag{3.2}$$

Then as the theory assumed the existence (or non-existence) of a certain characteristic of a class has no influence to/from the existence (of non-existence) of any other characteristic. When for every $\mathrm{F_i}$ is conditional independent of every $\mathrm{F_j}$ for $\mathrm{i} \neq \mathrm{j}$, we have:

$$\mathrm{p}\left(\mathrm{F_i} \middle| \mathrm{C}, \mathrm{F_j}\right) = \mathrm{p}(\mathrm{F_i}\ |\mathrm{C}) \tag{3.3}$$

So, the joint model of the numerator can be changed to:

$$\mathrm{p}(\mathrm{C}, \mathrm{F_1}, \dots, \mathrm{F_n}) = \mathrm{p}(\mathrm{C}) \cdot \prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{p}(\mathrm{F_i}|\mathrm{C}) \tag{3.4}$$

We could then rewrite the expression:

$$\mathrm{p}(\mathrm{C}|\mathrm{F_1}, \mathrm{F_2}, \dots, \mathrm{F_n}) = \frac{1}{\mathrm{p}(\mathrm{F_1}, \dots, \mathrm{F_n})} \mathrm{p}(\mathrm{C}) \prod_{\mathrm{i}=1}^{\mathrm{n}} \mathrm{p}(\mathrm{F_i}|\mathrm{C}) \tag{3.5}$$

3.1.2 Learning progress

As we discussion before, either E-mail spam or forum spam are both linguistic expressions. So, in this case, we are facing an object which is post P and features of a post which are words w. If there are only two classes, spam post and ham post, we want to classify each of the posts into either spam post S or ham post $\overline{\mathrm{S}}$.

If we use Eq. (3.6) to denote the probability that the $\mathrm{i^{th}}$ word of a given post occurs in a post from class C, then in the expression (3.7) we have the probability of a post P given a class C:

$$\mathrm{p}(\mathrm{w_i}|\mathrm{C}) \tag{3.6}$$

$$\mathrm{p}(\mathrm{P}|\mathrm{C}) = \prod_{\mathrm{i}} \mathrm{p}(\mathrm{w_i}|\mathrm{C}) \tag{3.7}$$

Now, can we get the probability of a given post P within a given class C?

What is $\mathrm{p}(\mathrm{C}|\mathrm{P})$? From Bayesian Theorem we could have the following equations

$$\mathrm{p}(\mathrm{P}|\mathrm{C}) = \frac{\mathrm{p}(\mathrm{P} \cap \mathrm{C})}{\mathrm{p}(\mathrm{C})} \tag{3.8}$$

$$\mathrm{p}(\mathrm{C}|\mathrm{P}) = \frac{\mathrm{p}(\mathrm{P} \cap \mathrm{C})}{\mathrm{p}(\mathrm{P})} \tag{3.9}$$

$$\mathrm{p}(\mathrm{C}|\mathrm{P}) = \frac{\mathrm{p}(\mathrm{C})}{\mathrm{p}(\mathrm{P})} \mathrm{p}(\mathrm{P}|\mathrm{C}) \tag{3.10}$$

Now we consider the two classes S and $\overline{\mathrm{S}}$ where every post belongs to either one or the other:

Giving

$$\mathrm{p}(\mathrm{P}|\mathrm{S}) = \prod_{\mathrm{i}} \mathrm{p}(\mathrm{w_i}|\mathrm{S}) \tag{3.11}$$

with

$$p(P|\bar{S}) = \prod_i p(w_i|\bar{S}) \tag{3.12}$$

Adopting the Bayesian result above, we have:

$$p(S|P) = \frac{p(S)}{p(P)} \prod_i p(w_i|S) \tag{3.13}$$

$$p(\bar{S}|P) = \frac{p(\bar{S})}{p(P)} \prod_i p(w_i|\bar{S}) \tag{3.14}$$

Then we divide the upper one by the lower one:

$$\frac{p(S|P)}{p(\bar{S}|P)} = \frac{p(S) \prod_i p(w_i|S)}{p(\bar{S}) \prod_i p(w_i|\bar{S})} \tag{3.15}$$

or we can write like this:

$$\frac{p(S|P)}{p(\bar{S}|P)} = \frac{p(S)}{p(\bar{S})} \prod_i \frac{p(w_i|S)}{p(w_i|\bar{S})} \tag{3.16}$$

Taking the logarithm of the above equation:

$$\ln \frac{p(S|P)}{p(\bar{S}|P)} = \ln \frac{p(S)}{p(\bar{S})} + \sum_i \ln \frac{p(w_i|S)}{p(w_i|\bar{S})} \tag{3.17}$$

3.1.3 Validation phase

In the end, the post can be categorized to spam if:

$$\ln \frac{p(S|P)}{p(\bar{S}|P)} > 0 \tag{3.18}$$

otherwise if the result from Eq. (3.18) is smaller than zero it’s a ham.

3.2 k Nearest-Neighbour

The k Nearest-Neighbour (k-NN) is a very simple approach used in documents classification and it also has a very good performance on the text categorization tasks. It’s known as a “lazy-learning” approach, which means it doesn’t need the learning phase in the recognition process. To accomplish the classification tasks, it only needs to create a vector space for each document category which will preserve all the information in the category. [11]

3.2.1 Vector space

To explain the vector space in this project, we use a simply example as follows. If one document contains only one sentence “This cat and this dog are both hungry.” then the feature vector of this document is [this, cat, and, dog, are, both, hungry] which include all terms in the documents without duplication. And as what we did in this feature vector, to reduce the unnecessary inaccuracy brought by different formats of terms, we will transfer all capital letters into small letters in the following experiments.

Then, in one document set, the vector space is a collection of all terms which occur in this document set without duplication. For documents in this document set, each of them will be associated to the unique feature vector which is generated from the vector space. What we explained above is that feature vector of each document only contains the terms occur in that document, however, for the purpose of Euclidean distance computing which we will explain later, this feature vector needs to be extended to the same length as the vector space of this document set. Each term in the feature vector of each document will be assigned a tf-idf weight.

3.2.2 Parameter *k*

The triangles, circles and rectangles showing in Figure 3-1 and Figure 3-2 denote three different categories of objects which are used as training data in the k-NN algorithm. The pentagram, in the centre of both pictures, is a category-unknown object which has the same vector space as other objects presented in the picture. And the thin round circle indicates the radius of a neighbourhood using pentagram as the centre.

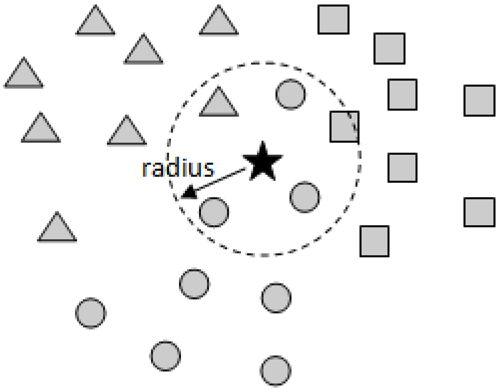


Figure 3-1 object with small neighbourhood

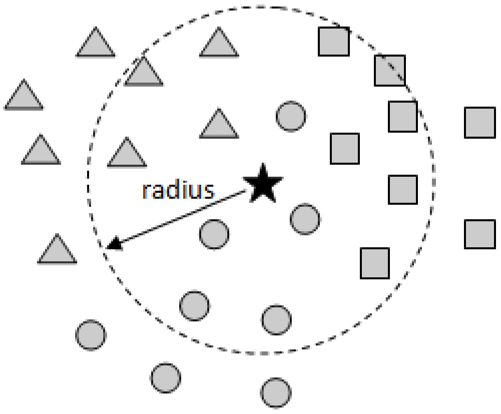


Figure 3-2 object with large neighbourhood

To classify this unknown object, we search the objects nearby this unknown one in a set radius and decide how many neighbours will be treated as the nearest ones. If in these k nearest neighbours there are m (m equal or less than k) objects in the same class ‘A’ and they are the majority in these k nearest neighbours, we could then classify this unknown object to class ‘A’ as well.

However, the parameter k couldn’t be chosen randomly and actually the k should be chosen very carefully, in other words, the parameter k may decide which class the unknown object will be placed in. We will explain the reason by using Figure 3-1 and Figure 3-2.

If we also count the objects which land on the border of the neighbour, we will get one triangle, three circles and one rectangle in the neighbourhood in Figure 3-1. Since the majority part of these neighbours is circle, so the pentagram, unknown object, will be classified to class circle.

Comparing to Figure 3-1, we enlarge the radius of neighbourhood in Figure 3-2 and the result is four triangles, five circles and six rectangles in the neighbourhood. Since the majority part of these neighbours is rectangle, the known object will be classified to class rectangle.

In the above example, we could clearly understand the importance of parameter k and the challenge of deciding it. In the following experiments, we tested four parameter k and explained the result of these different choices.

3.2.3 Distance between documents

But before the selection of parameter k, we need to know who is in the neighbourhood and how far our algorithm can reach them. In [11], researchers use Euclidean distance and the angle between two documents to measure the similarity of two documents and then use the similarity as the radius of neighbourhood.

In this project, we also employ Euclidean distance to measure the distance between two documents. And the attribute we use to measure the Euclidean distance from equation (2.5) is feature vector of each document.

3.3 Self-organizing map

The self-organizing map (SOM), as an unsupervised machine learning method, is an effective tool for the visualization of high-dimensional data. It implements an orderly mapping of a high-dimensional distribution onto a regular low-dimensional grid. The map is used to convert complex, nonlinear statistical relationships between high-dimensional data objects into simple geometric relationships on a low-dimensional display. [12]

In this project, each document in the document set can be treated as a high-dimensional data. As we explained in section 3.2.1, the vector space, which may consist of hundreds of thousands of terms, used in k-NN algorithm will also be used in SOM. By utilizing SOM, the low-dimensional grid will preserve the most important topological and metric relationships of documents on the display and it may also give an abstraction of the document set.

3.3.1 Map construction & establishment

In this project, we will employ 2D grid to establish the SOM as demonstrating in the left side of Figure 3-3. In this grid, each dot denotes a node in the map which has a weight vector and the length of the weight vectors is as the same length as the input vectors. The input vectors, which are grouped in the right side of Figure 3-3, are generated from each document.

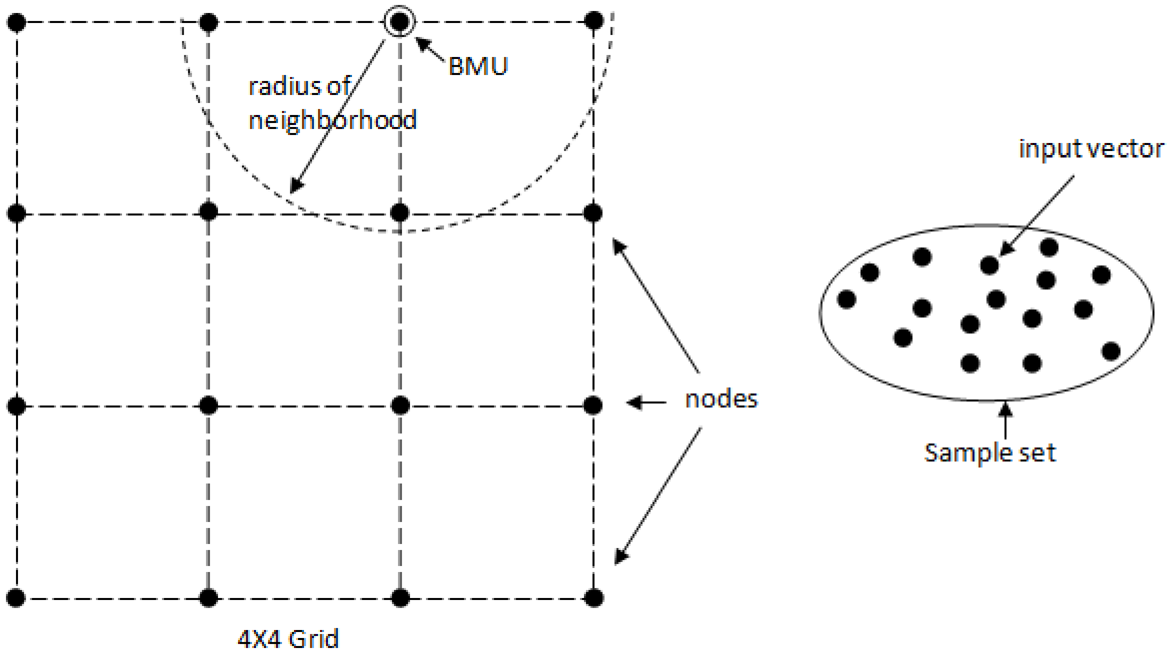


Figure 3-3 SOM demonstration

Generally, the establishment and training phase of this map may have several steps as follows:

1. Initialize the map by assigning each node a weight vector. The node vector length is as the same length as the input vector. (Usually the value of each weight is set between 0 and 1.)
2. Randomly select a sample vector from the input set.
3. Find the node in the map which has the smallest distance with the sample vector in “step 2” and it is referred to be the Best Matching Unit (BMU).
4. Using the current BMU as a centre of neighborhood, adapt each node’s weight vector within the neighborhood including BMU itself under a learning rate. The adaptations will make nodes be more similar to the sample vector.
5. After the adaptation, reduce the radius of neighborhood, as well as the learning rate.
6. Repeat from “step 2” to “step 5”.

The purpose we adapt nodes in the neighbourhood is to make their weight vectors move closer to the sample vector. After several learning iterations, nodes which have similar weight vectors indicating that they have the similar input patterns will be grouped specially close to one another. In [13] Dr. Dieter Merkl described a concrete adaptation approach which worked quite well in his research and we will utilize this approach in this project.

Each of the nodes i is assigned a weight vector m_i. During each learning iteration, the node c having the highest activity associated to a randomly chosen sample vector x is determined which is further referred to as the best matching unit (BMU). To measure the activity level of a unit, we use the common way that is to label the unit with Euclidean distance between its weight vector and the sample vector from sample set. Then, the selection of BMU c will be written as given in Eq. (3.19) with $||\blacksquare||$ denoting the Euclidean vector norm. In this and the following expressions, we make use of a discrete time notation with t denoting the current training iteration:

$$\mathrm{c}: \|\mathrm{x}(\mathrm{t}) - \mathrm{m_c}(\mathrm{t})\| = \mathrm{min_i}(\|\mathrm{x}(\mathrm{t}) - \mathrm{m_i}(\mathrm{t})\|) \tag{3.19}$$

Adaptations will be carried out at each of learning iterations which is known as a steady reduction of the difference between unit weight vector and input sample vector. The amount of adaptations is ruled by integrating a learning rate $\propto(\mathrm{t})$ which steady reduces in the process of training phase. Under the steady reductive learning rate, large adaptations will be carried out at the beginning of training phase where the randomly initialized unit's weight vector has to be modified towards the input vector space. At the last few small adaptations, learning behaviours towards the fine-adjustment from input vectors.

The sample scenario in Figure 3-3 above, we use the circled node in the grid to denote a BMU which is calculated when a random sample vector form sample set is presented to the map. As well as the learning rate, the neighbourhood of BMU will be steady decreasing at each of learning iteration and the weights of all units within the neighbourhood of BMU are changing according to the iterations. The trend of adaptation which those units in the neighbourhood follow is as the same as the adaptation of BMU.

Units in the neighbourhood of the physical location of BMU are submitted to adaptation under the consideration of current unit i and BMU c taking a neighbourhood function $\mathrm{h_{c(x),i}}$ into account. In this adaptations, units closer to the BMU will be effected more strongly than those further away. The Eq. (3.7) describes the reductive neighbourhood of BMU with $\mathrm{r_i}$ denoting the node's physical location in grid which is a two dimensional vector and $\|\mathrm{r_i} - \mathrm{r_c}\|$ denoting the distance between unit i and c.

$$\mathrm{h_{c(x),i}}(\mathrm{t}) = \mathrm{e}^{-\|\mathrm{r_i}-\mathrm{r_c}\|^2/2\sigma^2(\mathrm{t})} \tag{3.20}$$

Commonly, the range of neighbour shrinks from a large value and the spatial range of units processed in adaptations is decreased steady during the training phase. The parameter σ in Eq. (3.7) is the time-varying parameter.

According to the above description for SOM, we present the adaptation formula here:

$$\mathrm{m_i}(\mathrm{t}+1) = \mathrm{m_i}(\mathrm{t}) + \alpha(\mathrm{t}) \cdot \mathrm{h_{c(x),i}}(\mathrm{t}) \cdot [\mathrm{x}(\mathrm{t}) - \mathrm{m_i}(\mathrm{t})] \tag{3.21}$$

3.3.2 Classification

The task of utilizing SOM in this project aims not only to cluster different documents but also to classify new documents into those clusters in the future. In this validation phase, we will use the information obtained from SOM to carry out classification process.

From the above processes, we have already gained the SOM with each unit having a weight vector with it. This weight vector formed processing vast amount of input data will have some kind of feature abstraction or topology of input vector space.

Since those features in each unit associating with values, we could simply use the features with high value as eigenvector of this unit. Based on these eigenvectors, we manually assign each unit to a category in the map. Therefore a visualized distribution map of categories can be employed as a document classifier.

During the classification phase, we simply calculate the distance between weight vector of each unit and feature vector of candidate data. The unit which is the closest to the candidate data will mark the candidate data with the category in which this unit is.

Chapter 4

Prototypes

The prototype, served as a proof-of- concept endeavour, is developed to demonstrate either our assumptions with spam classification perform well or not.

Theories, in other words formulas and methods, described in the previous chapters will be translated into algorithms and implemented by Python programming language and supporting frameworks.

We totally prepared more than 3000 posts separated in training set and test set. Training set contains more than 2000 posts in ham set and spam set respectively while test set has 40 threads with the number of posts in each thread various from less than 10 to more than 60.

The following sections will describe our implementation in detail as well as some results. And those discoveries will be further discussed in the following Chapter 5.

4.1 Data set

To make our classifier fit in the real-world scenario, we used two groups of data, training group and test group, and each group contains both ham posts and spam posts. Besides, to challenge our classifier, the domain in which the posts in the training set stay is different from the domain in which the posts in the test set stay.

Training group: There are two sets of posts in this group, ham set and spam set, with around 1000 pure ham posts and 1000 spam posts collections in each set. Posts in ham set are most in the domain which is about mobile in UK forums and posts in spam set are forum advertisement from forums in UK and non-UK but using English as the communication language.

Test group: There are 40 threads of posts in this group and each of them is an independent scenario with its own topic which is about movies, phones etc. In this case, candidate data contains unfamiliar information with training group. Furthermore, spam posts are inserted in some of these scenarios and the opinion of doing this intends to increase the difficulty of classification. Since in the real forums, spammers will not insert many spam posts in one thread, we didn't insert more than 3 spam posts in each thread. The intention of doing this is to try to make the test group look similar to real threads in forums.

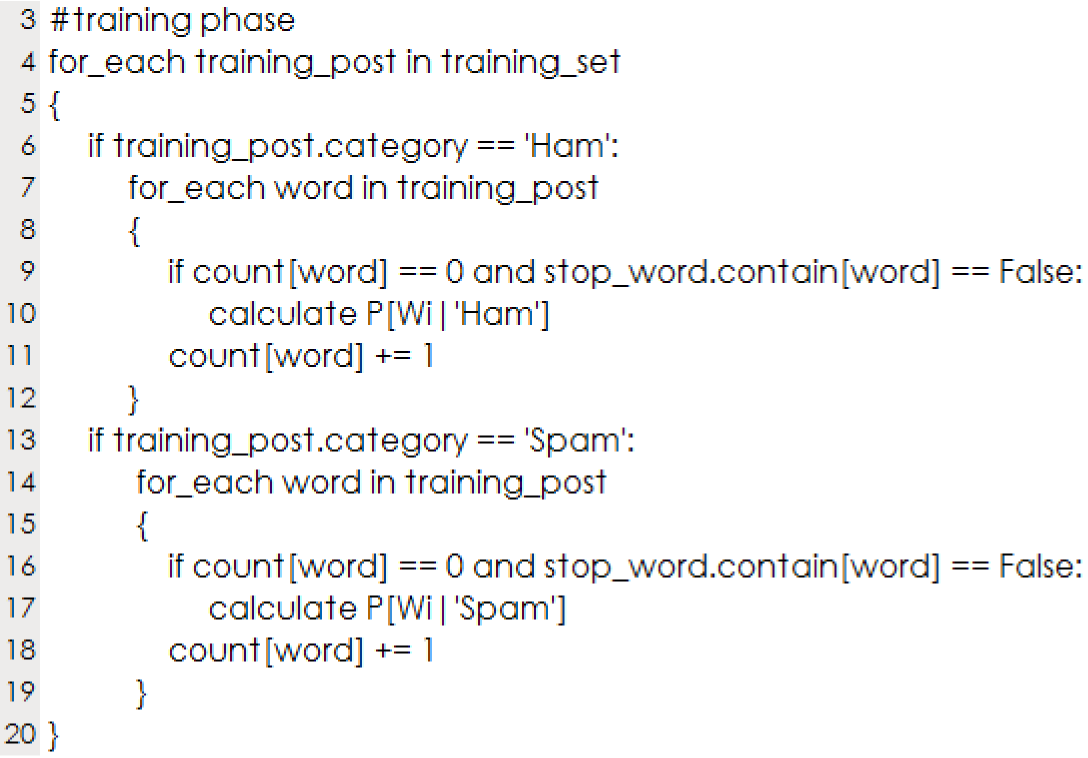
4.2 Naïve Bayes combining with Novelty detection

Naïve Bayes classifier is a widely used text classification theory in anti-spam field, especially applied to E-mail spam filtering. The purpose of implementing Naïve Bayes in this project is to verify this theory could also be used in SMC. To help approving this assumption, we prepared two sets of data which are training set and test set and the contents of these two sets are clearly described above, so we do not spend time to repeat it.

4.2.1 Pure Naïve Bayes classifier

As a supervised machine learning algorithm, Naïve Bayes classifier needs two phases, training phase and validation phase, to carry out the classification. We will use data from training group in the training phase and data from test group in the validation phase.

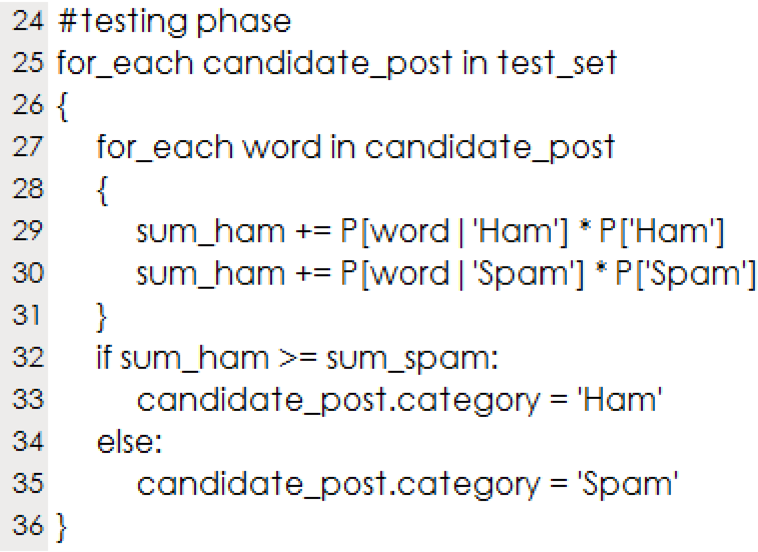
The training phase is carried out by the book. To get the probability $p(w_i|C)$ from Eq. (3.6), we need to get a word list for each post. To strip away some high frequency common words like: “and”, “I”, “my”, “if”, “you”, etc, we attach a stop words document with the classifier which is offered by Prof Ole-Christoffer Granmo. Code 4-1 shows the training phase of Naïve Bayes.



```
#training phase
for_each training_post in training_set
{
    if training_post.category == 'Ham':
        for_each word in training_post
        {
            if count[word] == 0 and stop_word.contain[word] == False:
                calculate P[Wi | 'Ham']
            count[word] += 1
        }
    if training_post.category == 'Spam':
        for_each word in training_post
        {
            if count[word] == 0 and stop_word.contain[word] == False:
                calculate P[Wi | 'Spam']
            count[word] += 1
        }
}
```

Code 4-1

The validation phase checks all terms in the candidate post and calculate the maximum possible category for each term which is in Code 4-2. Based on the probability of classes given by each term, the classifier will choose the most likely class for the candidate post.



```
#testing phase
for_each candidate_post in test_set
{
    for_each word in candidate_post
    {
        sum_ham += P[word | 'Ham'] * P['Ham']
        sum_ham += P[word | 'Spam'] * P['Spam']
    }
    if sum_ham >= sum_spam:
        candidate_post.category = 'Ham'
    else:
        candidate_post.category = 'Spam'
}
```

Code 4-2

The result of classification for pure Naïve Bayes classifier is given in Table 4-1:

Naïve Bayes	Ham	Spam	Accuracy
Ham	1062	112	90.5%
Spam	0	21	100.0%

Table 4-1 pure Naïve Bayes classification result

4.2.2 Enhanced Naïve Bayes classifier

The expectation of classifications will be 100% right but there are still some wrong categorization results from the pure Naïve Bayes classifier. To enhance it without additional information, we choose to use the Novelty detection method which only needs the data in the test set.

In fact, the Novelty detection is developed to track the new and unknown part of data from a data retrieval system for users. But here, we use it to track the unfamiliar part based on the topic of a certain discussion.

We actually implemented two approaches to integrate novelty detection with Naïve Bayes: the first one is to only use weight of each post and then make the decision of corrections based on those values; the second is to calculate cosine similarity, based on the weight of each post, between posts from which we could decide the direction to correct Naïve Bayes classifier.

First approach: To accomplish the first task, we need to assign a weight for each post and this weight should be reasonable and easy to obtain. We use normalized tf-idf value as the weight for each post and then calculate the mean value based on weights.

In the calculation phase, we encountered a theory flaw. Based on equation (2.2), we sometimes got '0' result for $\mathrm{idf_i}$ value due to the calculation of $\frac{|D|}{|\{d:t_i \in d\}|}$ is sometimes occasional to be '1'. Follow the idea in [14], we changed the tf and idf calculation formulas to:

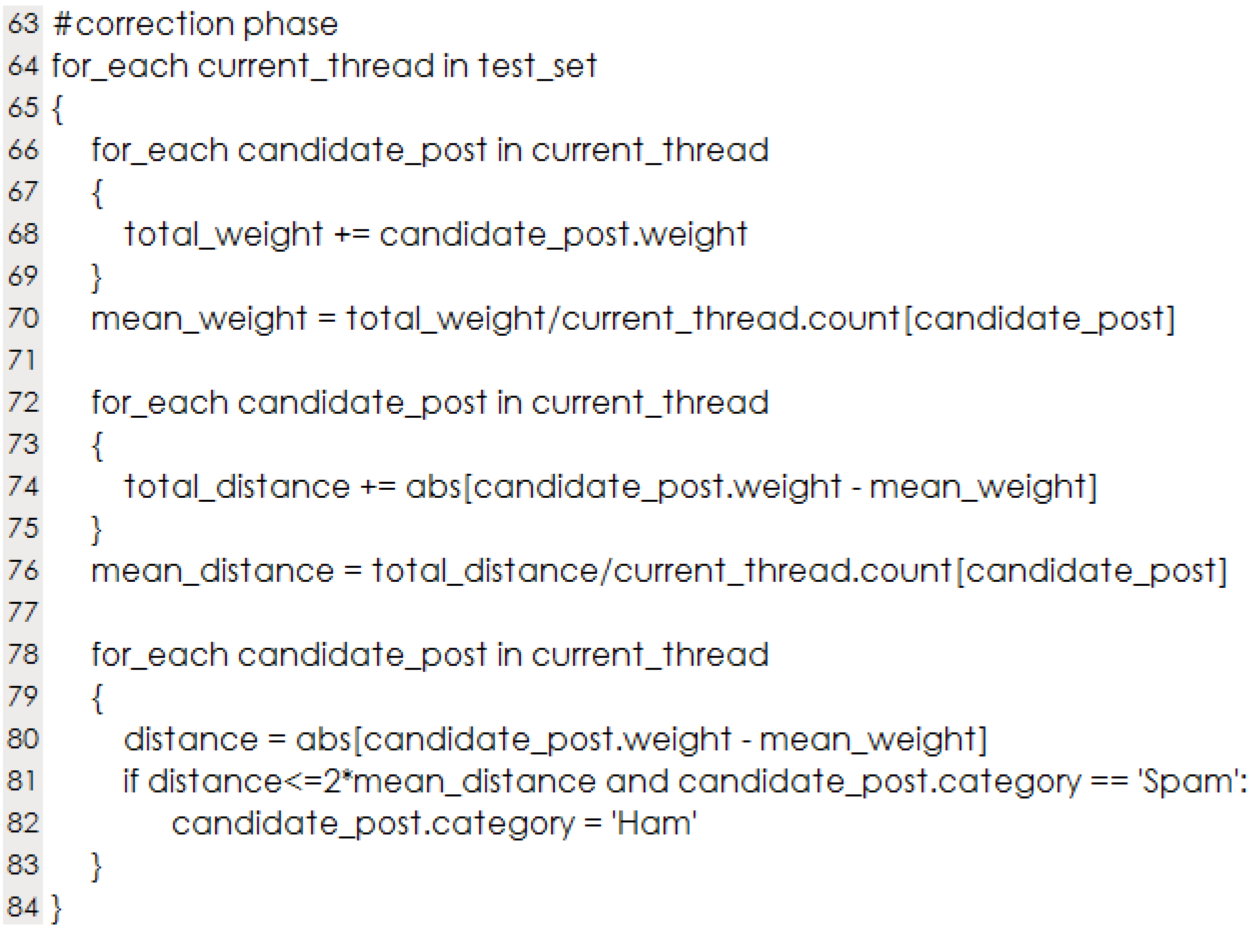
$$\mathrm{tf} = \log(\mathrm{term_{frequency}} \ + \ 1.0) \tag{4.1}$$

$$\log((\mathrm{Doc_{count}} \ + \ 1.0)/(\mathrm{Doc_{frequency}} \ + \ 0.5)) \tag{4.2}$$

After obtaining those weights, we need to choose a parameter which is used to be a boundary to separate spam and ham. With the help of supervisor Jaran, we decided to use the value called double-mean that obtained as the following steps:

1. Calculate the mean value of all weights in the thread
2. Calculate the distance between each weight and the mean value obtained in step '1'
3. Calculate the mean value of all distances obtained in step '2'
4. Multiply the mean value obtained in step '3' by 2

The double-mean value from step '4' will be chosen as a boundary between two categories. The pseudo Code 4-3 gives the validation phase based on both results from Naïve Bayes classifier and Novelty detection by using only weights:



```
#correction phase
for_each current_thread in test_set
{
    for_each candidate_post in current_thread
    {
        total_weight += candidate_post.weight
    }
    mean_weight = total_weight/current_thread.count[candidate_post]

    for_each candidate_post in current_thread
    {
        total_distance += abs[candidate_post.weight - mean_weight]
    }
    mean_distance = total_distance/current_thread.count[candidate_post]

    for_each candidate_post in current_thread
    {
        distance = abs[candidate_post.weight - mean_weight]
        if distance<=2*mean_distance and candidate_post.category == 'Spam':
            candidate_post.category = 'Ham'
    }
}
```

Code 4-3

The result of classification for enhanced Naïve Bayes classifier with Novelty detection by only using weights is given in Table 4-2:

Naïve Bayes & Novelty detection	First approach	Ham	Spam	Accuracy
	Ham	1163	11	99.1%
	Spam	5	16	76.2%

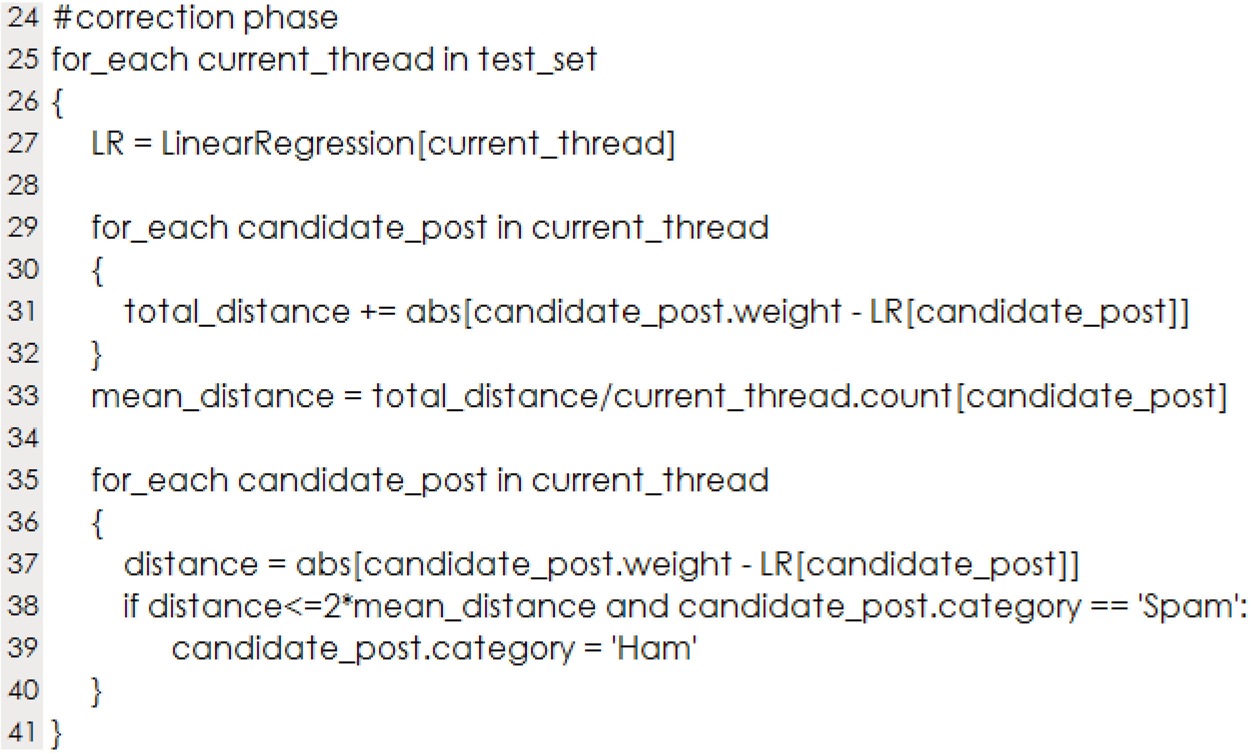
Table 4-2 Naïve Bayes combining with Novelty detection first approach classification result

Second approach: To accomplish the second task, we need to use some results from the implementation of the first approach. We use the weight for each post calculated in the first task as our source to compute cosine similarity between each pair of posts.

Then any post in the thread will have a series of cosine distances between it and others and we sum them together as the cosine similarity value assigned to each post. By doing this, there will be an overview that how far a post will be away from others in total.

The cosine similarity curve we plot in MATLAB shows a strong descending trend in those last few points which inspire us to try to test how far those points will be away from the linear regression line of similarity values.

The pseudo Code 4-4 gives the validation phase based on both results from Naïve Bayes classifier and Novelty detection by using cosine similarity:



```
#correction phase
for_each current_thread in test_set
{
    LR = LinearRegression[current_thread]

    for_each candidate_post in current_thread
    {
        total_distance += abs[candidate_post.weight - LR[candidate_post]]
    }
    mean_distance = total_distance/current_thread.count[candidate_post]

    for_each candidate_post in current_thread
    {
        distance = abs[candidate_post.weight - LR[candidate_post]]
        if distance<=2*mean_distance and candidate_post.category == 'Spam':
            candidate_post.category = 'Ham'
    }
}
```

Code 4-4

The result of classification for enhanced Naïve Bayes classifier with Novelty detection by using cosine similarity is given in Table 4-3:

Naïve Bayes & Novelty detection	Second approach	Ham	Spam	Accuracy
	Ham	1161	13	98.9%
	Spam	20	1	4.8%

Table 4-3 Naïve Bayes combining with Novelty detection second approach classification result

4.3 k Nearest-Neighbour combining with Novelty detection

In the previous chapter, we discussed the k Nearest-Neighbour algorithm in a theoretical way and used the phrase “lazy-learning” to describe it. In this section, the implementation of k-NN below may be made use of as an interpretation of the concept that explains the algorithm with programming language Python.

Data prepared from above will be also used in this implementation. Since in this algorithm we need to obtain distances between each post in the test set and training set, we will use massive calculation in the prototype.

4.3.1 Pure k Nearest-Neighbour classifier

The k-NN classifier includes of two phases, training phase and validation phase. During the training phase, we need to assign each post a feature vector which consists all terms in that post. Before we can assign each post the feature vector, there are several preparations.

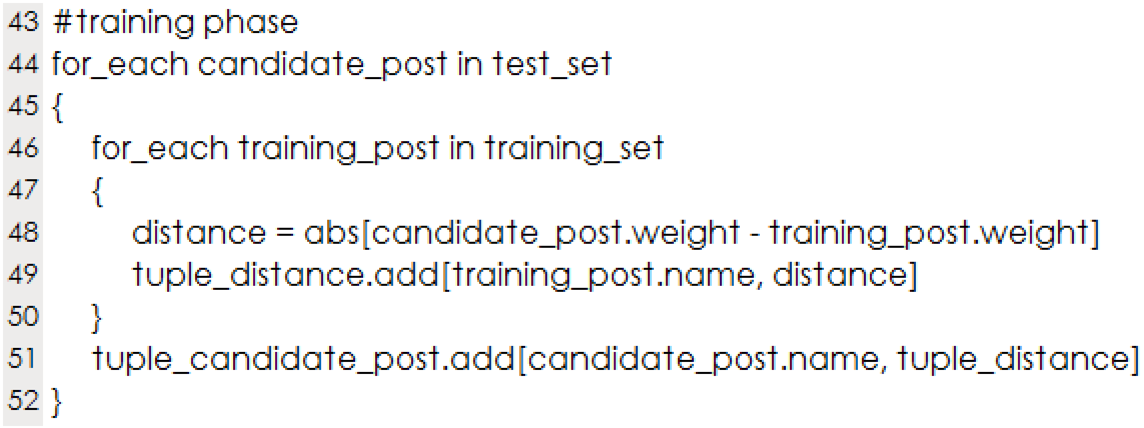
In the Naïve Bayes classifier, we already have implemented the calculation function to obtain tf-idf values for each post so we do not spend time to repeat it.

Each post, as we described, will have a feature vector which is a list of tuples having term and tf-idf weight with it. To calculate Euclidean distance between two posts, not only the number of tuples but also the term in each tuple has to be the same. For example, if post A has the feature vector {(work, 0.001), (job, 0.002), (carrier, 0.003)} and post B has the feature vector {(work, 0.003), (job, 0.01)}, in order to calculate the distance between them, we need to pad a tuple (carrier, 0) to the feature vector belonging to post B.

Assuming we have enough space to store all terms occurring in our data sets, we could then establish a huge vector space which contains all terms and assign this gigantic vector to each post. In this way, each post will have a complete list of all terms with a fixed order that will bring a lot of convenience when we need to calculate Euclidean distances.

However, this method will waste a lot of memory which is not what we can afford. So, to one post, we only store a small list of tuples which consists of terms occurring in this post with their tf-idf weight. If we encounter different number of tuples or different terms in tuples when calculate distances between posts, we need firstly to make two feature vectors have exactly the same length and contain exactly the same terms and for those terms having no occurrence in that post we will assign a empty value to them for the convenience of calculation.

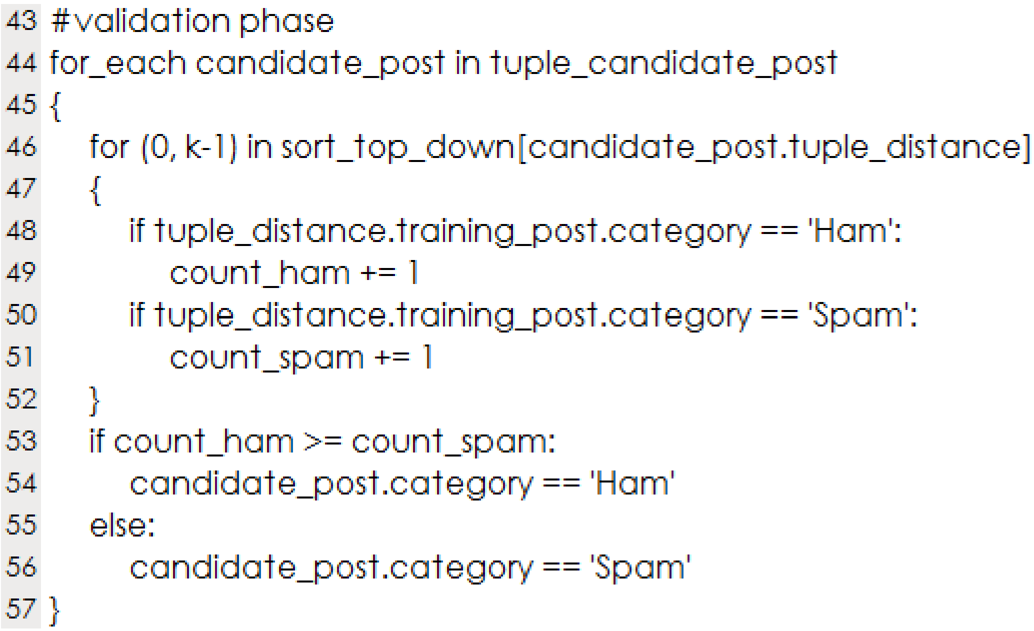
The following pseudo code simply presents the idea of k-NN classifier which also contains two phases. In Code 4-5, we see that the classifier will store a list with tuples which consist of name of the current candidate post and the distance between this post and every post in the training set.



```
#training phase
for_each candidate_post in test_set
{
    for_each training_post in training_set
    {
        distance = abs[candidate_post.weight - training_post.weight]
        tuple_distance.add[training_post.name, distance]
    }
    tuple_candidate_post.add[candidate_post.name, tuple_distance]
}
```

Code 4-5

The following Code 4-6 is our validation phase of k-NN classifier. During this validation phase, users can define the parameter k by themselves. Line 21 represents the process to choose the k closest neighbours which have the shortest distance between candidate post and training posts. In this k nearest neighbours, if posts belong to 'Ham' class more than posts belong to 'Spam' class, the categorization result will be 'Ham', vice versa.



```
43 #validation phase
44 for_each candidate_post in tuple_candidate_post
45 {
46     for (0, k-1) in sort_top_down[candidate_post.tuple_distance]
47     {
48         if tuple_distance.training_post.category == 'Ham':
49             count_ham += 1
50         if tuple_distance.training_post.category == 'Spam':
51             count_spam += 1
52     }
53     if count_ham >= count_spam:
54         candidate_post.category == 'Ham'
55     else:
56         candidate_post.category == 'Spam'
57 }
```

Code 4-6

The classification results showing in the following Table 4-4 are results from our pure k-NN classifier testing with four k selections.

k Nearest-Neighbour		Ham	Spam	Accuracy
k = 5	Ham	1080	94	92.0%
	Spam	8	13	61.9%
k = 10	Ham	1165	9	99.2%
	Spam	19	2	9.5%
k = 50	Ham	23	1151	2.0%
	Spam	0	21	100.0%
k = 100	Ham	1	1173	0.01%
	Spam	0	21	100.0%

Table 4-4 k-NN classification results

4.3.2 Enhanced k-NN classifier

From the above tests, when we select k to 50 or 100, we obtain quite interesting results. Even the classifier isn't sensitive enough to ham posts, we still get quite good result in the other direction which is that we have quite promising classification accuracy in spam classification. As for the accuracy of spam classification of our k-NN classifier when selection of k is between 5 and 10, we think there is enough room to improve it.

The approach to integrate Novelty detection is clearly described in the previous sections, so we just place the classification results in Table 4-5. After integrating the first approach of Novelty detection method with k-NN classifier, the accuracy of ham classification climbs straight up while the accuracy of spam classification declines somewhat.

k Nearest-Neighbour with Novelty detection		Ham	Spam	Accuracy
k = 5	Ham	1172	2	99.8%
	Spam	19	2	9.5%
k = 10	Ham	1172	2	99.8%
	Spam	20	1	4.8%
k = 50	Ham	1080	94	92.0%
	Spam	8	13	61.9%
k = 100	Ham	1078	96	91.8%
	Spam	8	13	61.9%

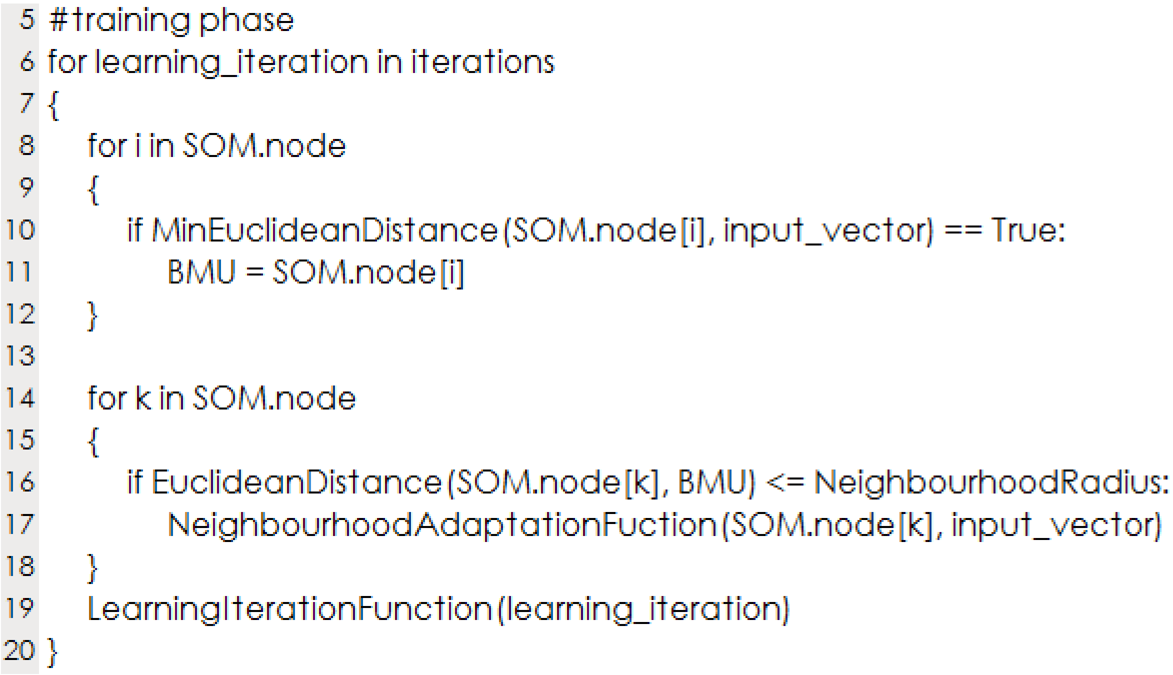
Table 4-5 k-NN combining with Novelty detection classification results

4.4 SOM classifier combining with Novelty detection

As an unsupervised machine learning algorithm, Self-Organizing Map is able to visualize high-dimensional data into low-dimensional grid. In order to map vast amount of data from data set, we firstly create a 2-dimensional grid with each node has the same vector length as the input vector.

4.4.1 Pure SOM classifier

Before the learning iterations, we firstly initialize the map with each node having a weight vector and the value of each weight is between zero and one. In each of the learning iterations showing in Code 4-7, we try to find a node which has the smallest Euclidean distance to the input vector and treat it as the Best Matching Unit (BMU).



```
#training phase
for learning_iteration in iterations
{
    for i in SOM.node
    {
        if MinEuclideanDistance(SOM.node[i], input_vector) == True:
            BMU = SOM.node[i]
    }

    for k in SOM.node
    {
        if EuclideanDistance(SOM.node[k], BMU) <= NeighbourhoodRadius:
            NeighbourhoodAdaptationFuction(SOM.node[k], input_vector)
    }
    LearningIterationFunction(learning_iteration)
}
```

Code 4-7

After we found BMU, all nodes in the neighbourhood of BMU including BMU itself will be put in the adaptation function according to the input vector. In this step, each node around BMU in the neighbourhood radius will be modified close to input vector.

As we described in the previous sections, the vector space for SOM algorithm is list of tuples which have term and weight with it. Each input vector has this list of tuples and the weight of terms in input vectors is obtained from tf-idf calculations. To save memory, we do not use tuples in the learning iterations but use only the weights. Since all weights are ordered according to terms or following the vector space, we could only make use of scalars during the entire learning process.

The following Figure 4-1 is a term clustering map from SOM based on our data from training set. In this 10X10 map, each node contains ten terms and terms from four of the nodes are showing in frames. These terms in each frame, from top to the bottom, are ranked by the popularity in each node which means top rank terms get top weight values.

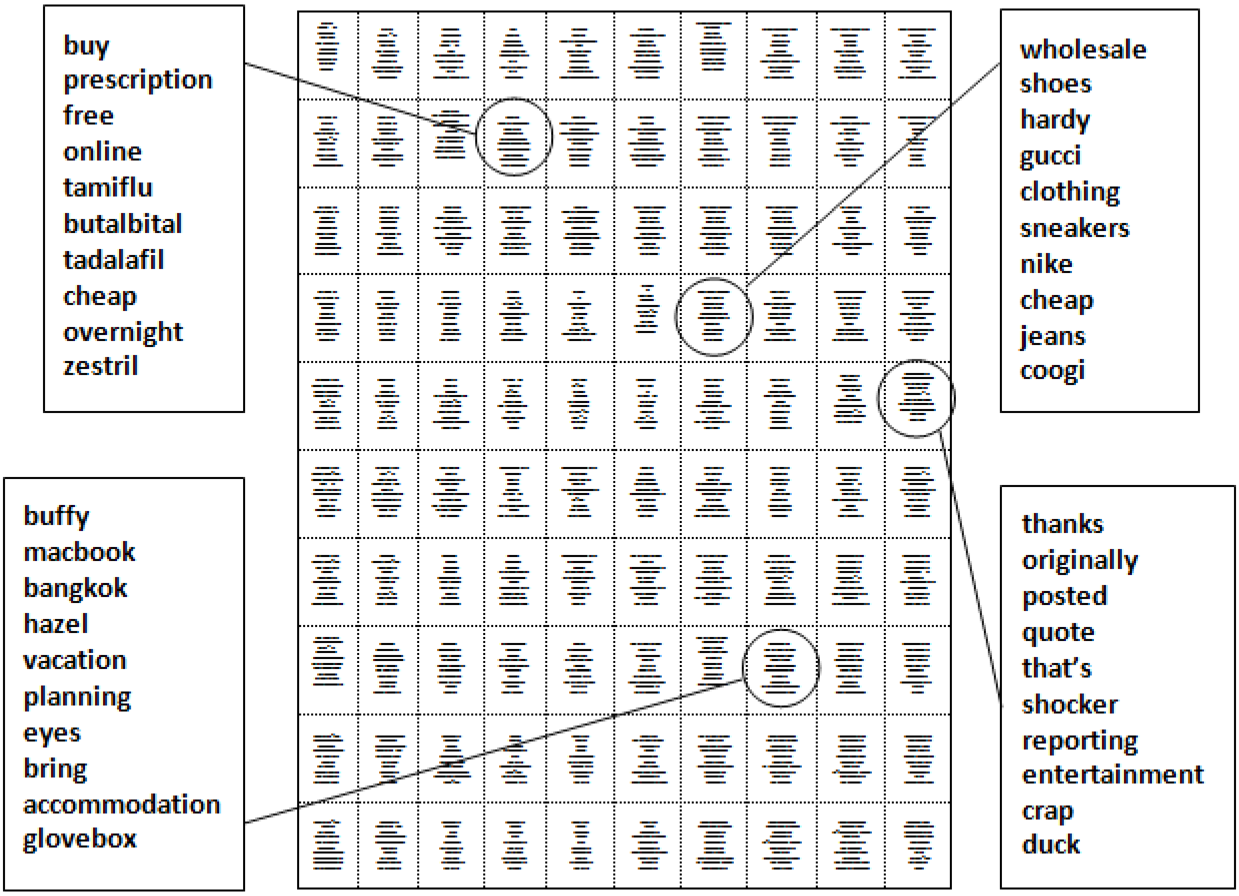


Figure 4-1 SOM node with high weight terms 1

Based on the terms in the above map, we could manually assign a category to each node. But the manual work takes time and may include man-made flaws. To reduce man-made mistakes and make our word more efficient, we integrate Naïve Bayes classifier with SOM classifier.

Clearly speaking, since we've already obtained the clustering map from SOM algorithm which is to say those top valued terms in each node contain the essential and abstract information of that node, we could use Naïve Bayes classifier to classifier each node just like what we did in the section 4-2 using training set as the training data.

The classification results is placed in Table 4-6 shows a very clear sign that some nodes are clustered to each other and terms in these node which are showed in Figure 4-1 are similar to each other.

Classification of each candidate post in the test set is based on the Table 4-6 when we make use of the SOM approach. The idea is quite clear. To classify each post from test set, we only need to find out which node has the smallest Euclidean distance to the post and then assign the post with the category label of this node.

Ham	Ham	Ham	Ham	Spam	Spam	Ham	Spam	Spam	Spam
Spam	Spam	Spam	Spam	Spam	Spam	Spam	Spam	Spam	Spam
Spam	Spam	Spam	Spam	Spam	Spam	Spam	Spam	Spam	Ham
Spam	Spam	Spam	Ham	Ham	Ham	Spam	Spam	Spam	Ham
Spam	Spam	Ham	Ham	Ham	Ham	Ham	Spam	Spam	Ham
Ham	Ham	Ham	Ham	Ham	Ham	Spam	Spam	Spam	Ham
Ham	Ham	Ham	Spam	Spam	Spam	Ham	Spam	Spam	Spam
Spam	Ham	Ham	Ham	Ham	Ham	Spam	Ham	Spam	Spam
Ham	Ham	Ham	Ham	Ham	Ham	Spam	Spam	Spam	Spam
Ham	Ham	Ham	Ham	Ham	Ham	Spam	Spam	Spam	Spam

Table 4-6 SOM node classification result 1

Here we have the classification result from SOM classifier in Table 4-7.

SOM	Ham	Spam	Accuracy
Ham	1150	24	98.0%
Spam	20	1	4.8%

Table 4-7 SOM classification result

4.4.2 SOM classifier with Novelty detection

To enhance the SOM classifier, we try to integrate Novelty detection with it. As we used in k-NN classifier, we also use the first approach of Novelty detection described in section 4.2.2 Enhanced Naïve Bayes classifier.

You may notice that the table in Figure 4-2 looks almost the same as the table in Figure 4-1 and the locations of four nodes which are chosen to demonstrate the results are as the same as the previous one in the map. However, terms in Figure4-2 in each node are not as the same as those in the previous table.

The reason is that we obtained this table after we re-ran the SOM classifier. From the algorithm, we can find the step 2 says: “Randomly select a sample vector from the input set.” Since each time, the input vector is randomly chosen from test set, we could imagine that the BMU in each of those iterations may be different. Not to mention that how different the results of SOM classifier will be after we re-ran the program with the total different initiation of the entire map.

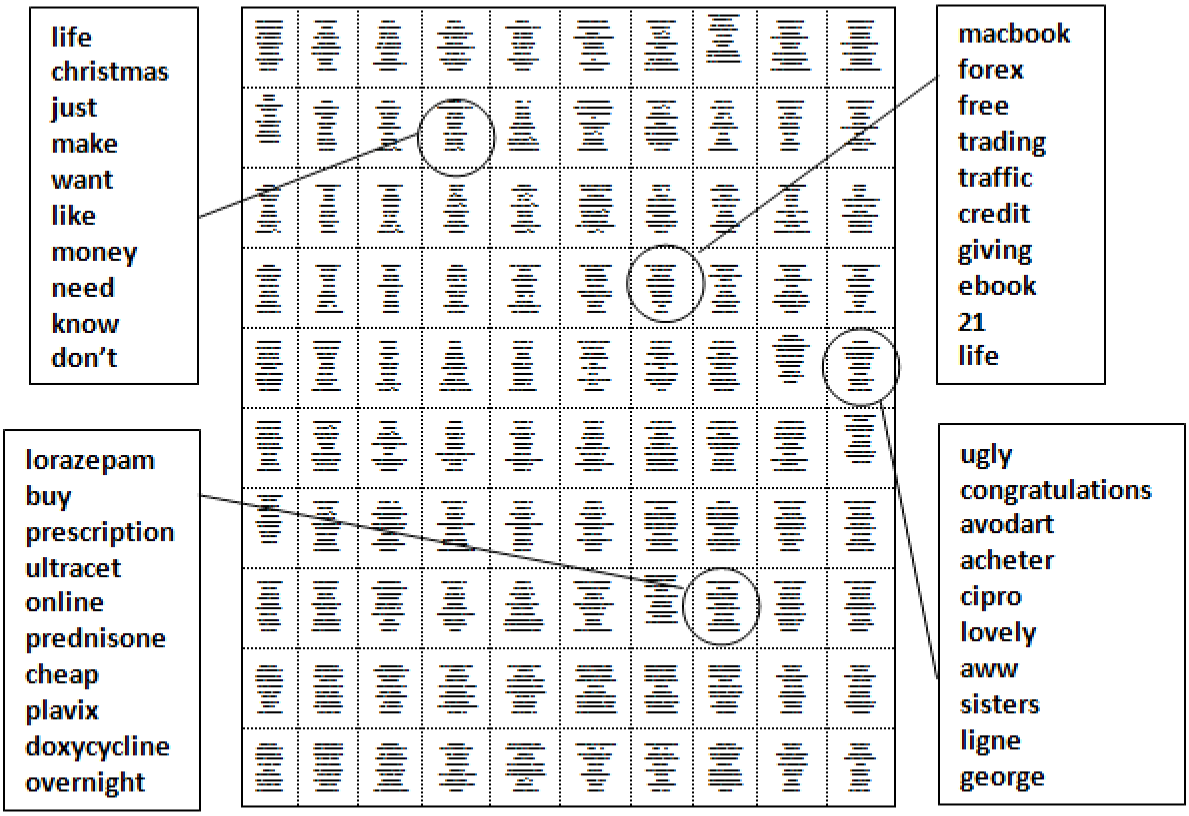


Figure 4-2 SOM *node with high weight terms 1*

Spam	Spam	Spam	Spam	Spam	Spam	Spam	Spam	Spam	Spam
Spam	Ham	Ham	Spam	Ham	Ham	Spam	Spam	Spam	Spam
Ham	Ham	Ham	Ham	Ham	Ham	Spam	Spam	Spam	Ham
Ham	Ham	Ham	Ham	Ham	Ham	Spam	Spam	Spam	Ham
Ham	Ham	Ham	Ham	Ham	Ham	Spam	Spam	Spam	Spam
Ham	Ham	Spam	Spam	Spam	Spam	Spam	Spam	Spam	Spam
Spam	Spam	Spam	Spam	Spam	Spam	Spam	Spam	Spam	Spam
Spam	Ham	Spam	Spam	Spam	Spam	Spam	Spam	Spam	Spam
Ham	Ham	Ham	Spam	Spam	Spam	Spam	Spam	Spam	Spam
Ham	Ham	Ham	Ham	Spam	Spam	Spam	Spam	Spam	Spam

Table 4-8 SOM node classification result 2

So does the classification of each node, having the different results from Table 4-6, which is based on those top weight terms in the Table 4-8 above. But, from the classification, we could still say similar nodes are clustered to each other since ‘Ham’ and ’Spam’ are clustered together respectively.

The classification result from enhanced SOM classifier is showing in Table 4-9.

SOM	Novelty detection	Ham	Spam	Accuracy
	Ham	1111	73	94.6%
	Spam	9	12	57.1%

Table 4-9 SOM classifier combining with Novelty detection classification result

Chapter 5 Discussion

In this chapter, we will deeply discuss each of those algorithms described in Chapter 3 and explain those results obtained from the prototypes in Chapter 4. To discuss them wisely, we will employ some graphs and statistics and this chapter will serve as a core discussion zone of challenges and difficulties in this thesis.

5.1 Novelty approach

Before we discuss the results obtained from each prototype, we firstly state again the intention of integrating Novelty approach with each prototype.

In Chapter 2, we described the intention to combine Novelty approach with other theories was to give a correction phase after each pattern classification process. Since each classifier will tread each post as an individual object, the classifier will not consider the environment where each individual post stays.

We assume some posts look meaningless or like spam individually and the fact is lots of posts in forums mean nothing if they don’t live with other posts in the same threads. Novelty approach will tread a whole thread as an entity or indivisible whole. So, each post in the thread either replies to the topic or replies to other posts will be meaningful treated by Novelty approach.

In this way, this approach gives us an opportunity to place the individual post back to its thread and recheck if it’s spam or not. So the decisions made by Novelty approach are also crucial to each of the classification processes.

5.2 Naïve Bayes classifier

In this section, we will compare performances of the pure Naïve Bayes classifier and Naïve Bayes classifier integrated with Novelty detection method. The well known Naïve Bayes classifier will not be the core topic in the following sections but the one with Novelty detection will be.

5.2.1 Good performance of Naïve Bayes

In our experiments, the ham classification accuracy of pure Naïve Bayes classifier reaches 90.5% and spam classification accuracy is perfect 100%. But the accuracy when it classifies ham posts is not good enough to us. In the previous project of course IKT 508, we did a better job where the Naïve Bayes classifier had reached the accuracy more than 98%. We'll try to explain the reason which causes the poor performance.

As what we explained in the previous chapters, the Naïve Bayes classifier makes use of supervised machine learning method. So, the performance of classification highly depends on the training data presented and the experiment data which will be classified.

In Chapter 4, we described that the ham posts in the training set are mainly collected from mobile or cell phone discussion sections and ham posts in the test set are collected from mobile, movie or TV shows discussions sections. The differences between these two sets are topics or to say it precisely the occurrences of high frequent common terms in these two sets are different.

Since training set has high frequent terms like 'phone', 'signal' or 'battery' but test set may have different high frequent terms such as 'star' or 'show', Naïve Bayes classifier learns from training set and may not consider 'star' or 'show' as high frequent ham terms. Under this condition, the classifier will make bad choice.

To improve the performance of Naïve Bayes classifier, we could change data from test set to data which is in the same domain as training data but we choose to leave it there and try to search a new approach.

5.2.2 Novelty approach

Figure 5-1 showing below is the screen shot of a straight line with all weights of posts in "Test set 25" and this test set will be used as a demonstration in the following discussions in this section.

Labelled dots are classified as spam by Naïve Bayes classifier, however, only the black circled one is a real spam. Due to the reason that those topics in which posts surround from our training set are mainly falling into the domain about mobile or cell phone discussions, however, in order to challenge our classifier we include discussions from other domains like movies or TV shows. Under this situation, Naïve Bayes classifier may give some wrong categorizations through the validation phase.

But, from this picture, we could tell that even thought one of those hams, at the right end of this line, is far away from each other, the classifier still works well to give the right classification which gives no doubt about the practicability of Naïve Bayes classifier.

Fortunately, we found out that by implementing the first approach of Novelty detection which uses only weights to correct the classification from Naïve Bayes that lots of wrong categorizations have been corrected. In Figure 5-1, the mean weight is ‘6.3412090839’ and the double-mean value we described above is ‘6.0178922508’ and then we could simply get the ham zone, where in this zone all posts which have been categorized in to ‘Spam’ will be re-categorized into ‘Ham’, is (‘0.323317’, ‘12.359101’). We could clearly see that the weight of the circled spam is 12.64 which is not within the ham zone.

Besides that, in Code 4-3, we tried to correct the ‘candidate_post.category’ to ‘Spam’ when the distance between post and mean weight of posts is larger than double mean distance, however, the results present that there are lots of right categorized ‘Ham’ by Bayes classifier will be wrongly categorized to ‘Spam’. We could confirm this by viewing the post in the end of the line in Figure 5-1 which is ham, but if we correct the post outside of the neighbourhood of the mean weight, it will be wrongly classified into ‘Spam’.

By receiving this kind of wrong corrections, which are caused by correcting posts outside of the neighbourhood of the mean weight, led us to go back to our samples and try to find the reason. After analyzing our candidate data combining with our classification results, we found out that some posters did post extraordinary posts or some posts contain too few words which is rare in the certain thread. But, if we look the thing from another angle, both of these reasons just proved the concept of novelty detection which is good at separating different ideas from ordinary documents.

In this sample, the Naïve Bayes classifier classified correctly on this circled spam but came along with several wrong categorizations. After integrated with Novelty detection approach, its behaviour of classifying ham is improved quite a lot.

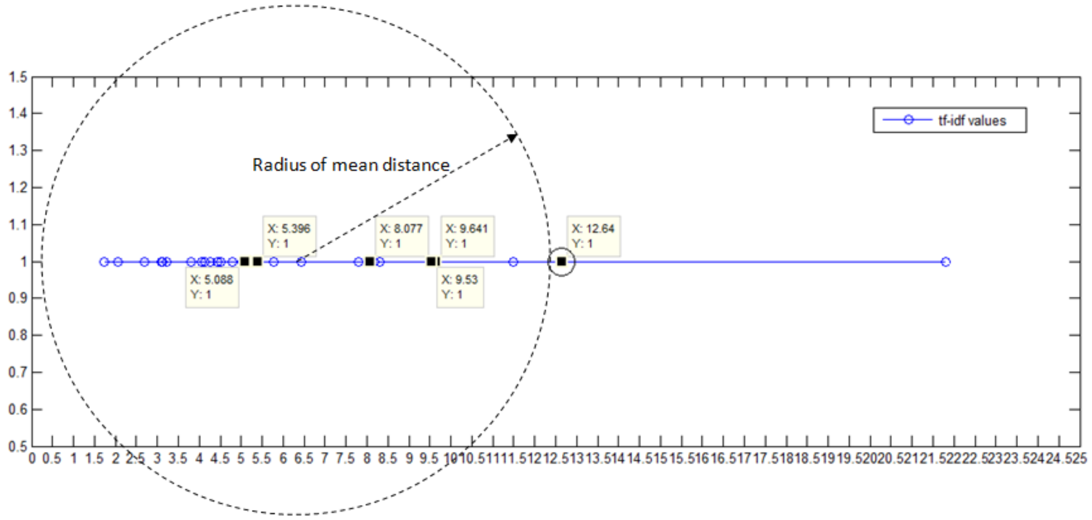


Figure 5-1 Naïve Bayes with Novelty detection experiment 1

Compare to just using weights of posts, the following implementation does even more work however gets worse results. From Table 4-3 we have seen that there is a slight improvement of categorizing ham from pure Naïve Bayes classifier but the categorizing of spam nearly fails to all.

The black circled dot in Figure 5-2 and Figure 5-3 is the same one in a different view. Figure 5-2 shows a flat view of cosine similarity values of the sample set and Figure 5-3 shows a curve view of cosine similarity with a linear regression line where we assign '1' to each weight in Figure 5-2 and in this way the straight could be expanded to a curve.

Here we present some results of these two pictures and then discuss why this solution is bad. The weight in the following discussion in section 5.1.2 refers to the weight calculated from second approach of Novelty detection.

In Figure 5-2, the mean value of these weights is '25.6738', the weight of spam is '25.5479' and the mean distance is '0.1615'. By a simple calculation we will obtain the distance between the spam and mean weight value is '0.1259' which is smaller than the average distance. In this situation, the novelty detection will give wrong correction which is to correct it into 'Ham'.

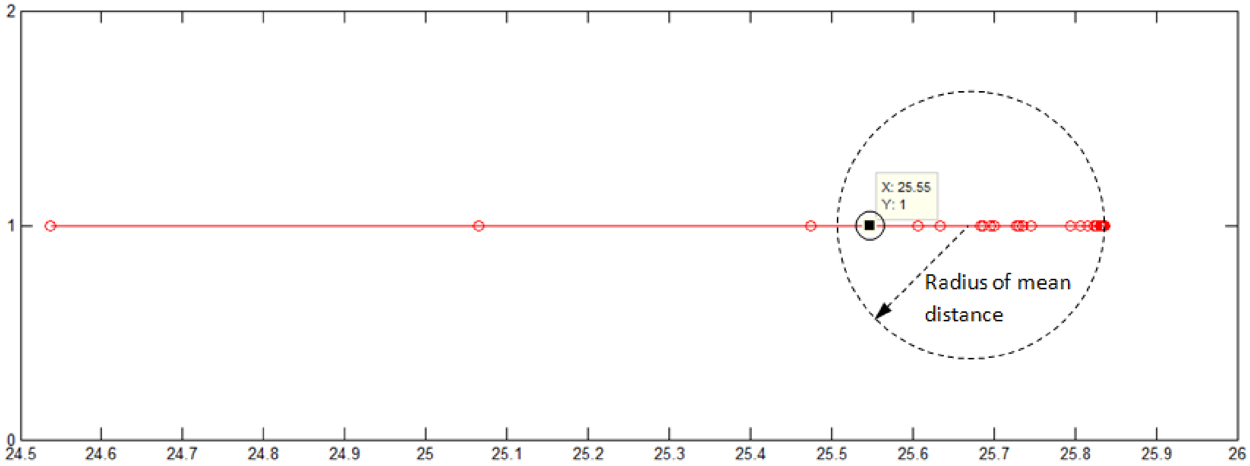


Figure 5-2 Naïve Bayes with Novelty detection experiment 2

A similar situation happens when we use the distance between cosine similarity values and linear regression values that the spam, the black circled one, appearing in the middle of this curve which is undistinguished from other hams.

The reason that post on the bottom of this curve generates so small cosine similarity value (the smaller the cosine distance is, the further the post will be away from others) is that this post has much less words comparing to other posts.

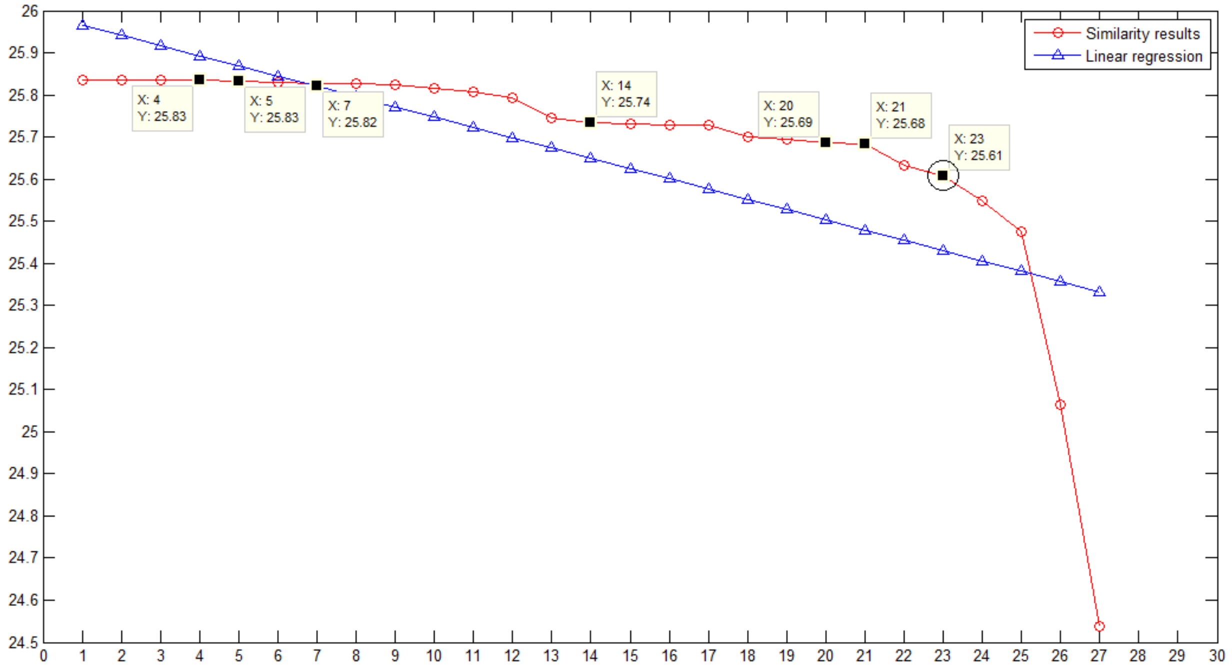


Figure 5-3 Naïve Bayes with Novelty detection experiment 3

This sample gives us a clear and brief image about why do we choose to use the first approach but not the second approach of Novelty detection.

5.2.3 Improvement

With Novelty detection, even the accuracy to classify spam posts drops from 100% to 76.2%, the accuracy of ham posts classification still rises from 90.5% to 99.1%. In this situation, we could say that the wrong classification of ham posts has been reduced a lot by integrating Novelty detection with it but it’s still not enough sensitive to spam posts.

Speaking to the improvements and deteriorations made by integrating Novelty detection with Naïve Bayes classifier, we think that the accuracy to classify ham posts can be treated as the same important as the accuracy to classify spam posts.

Because each normal discussion post in the thread is valuable to us with its value or potential value. To reduce the cost brought by information loss, it’s better to keep those undistinguished spam posts in the system rather than remove all spam posts and lots of ham posts mistakenly.

And by leaving some confusing spam posts in the classification results, we could also label them to distinguish them from ordinary posts and wait for further process. In this case, analysts may have not lost any information from the classification.

5.3 k-NN classifier

The sections next will bring us a thorough discussion about the performance of pure k-NN classifier and the k-NN classifier with Novelty detection. But let’s start from the pure k-NN classifier.

5.3.1 Pure k-NN classifier

The Figure 5-4 and Figure 5-5 are plotted by following Table 4-4 and Table 4-5 which are used to show the differences between two k-NN classifiers, one with Novelty approach and one without. Obviously to see, the results from k-NN classifier with Novelty detection is much better than pure k-NN classifier to classify ham posts when parameter k is selected to 50 and 100. But is that what we need?

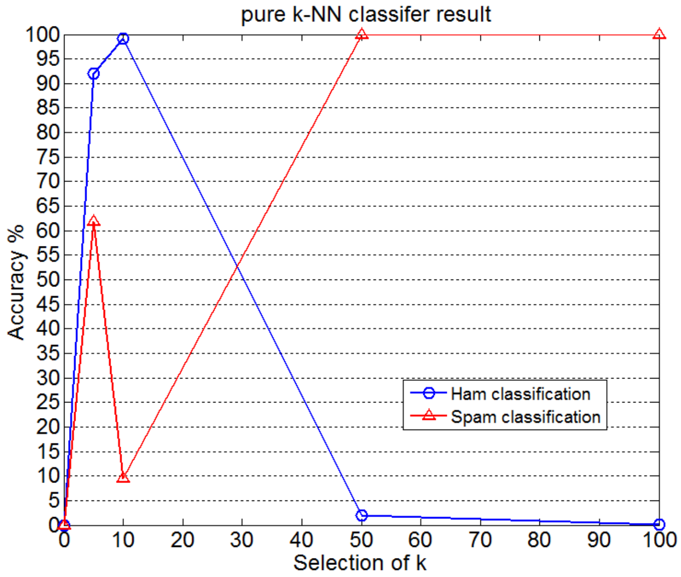


Figure 5-4 pure k-NN experiment

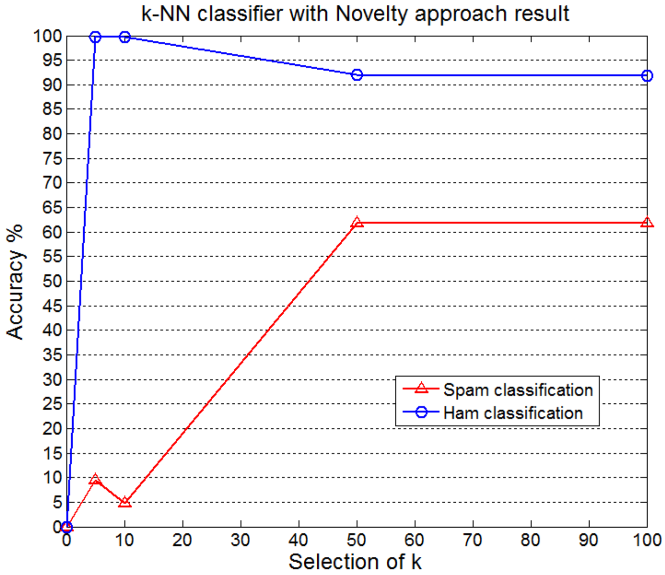


Figure 5-5 k-NN with Novelty detection experiment

Firstly we thoroughly discuss about the results from pure k-NN classifier and then compare them to the classifier with Novelty approach.

In the Figure 5-4, we have results from pure k-NN classifier. From Table 4-4, we know that the accuracy of classifying ham posts is 92.0% and the accuracy of classifying spam posts is 61.9% when we set k to 5. Comparing to pure Naïve Bayes classifier, the pure k-NN made less wrong classification on ham posts than pure Naïve Bayes classifier which is very promising news. On the other hand, the categorization of spam posts which is our main task isn't as good as the performance of pure Naïve Bayes.

When k grows to 10 with the same classifier, the accuracy of ham classification rises to 99.2% but the accuracy of spam classification drops to 9.5%. It's good to see the accuracy rising from 92% to 99.2% when use it to category ham, but it's quite disappointing to see the accuracy drops from 61.9% to 9.5% when it classifies spam.

But if we assume moderators or managers of forums will separate all posts being classified as spam from other normal forum posts, it will still save lots of work for them since there will be quite few wrong categorizations of ham posts.

Talking about the results from pure k-NN classifier when k is selected to be 50 or 100, we consider they are useless and may only be used as a compare group to demonstrate the importance of k selection. But when we combine the pure k-NN classifier with Novelty approach, it tells a totally different story.

5.3.2 A Novelty approach of k-NN classifier

The best part of our implementation of Novelty detection method is that it does not depend on training data. If we only read the statistics from Figure 5-5, we would say that it performs quite well with k-NN algorithm especially when k-NN classifier produces bad classification results.

However, when k is selected to 50 or 100, it takes more time and calculations to make decision but obtain the results when we just use the pure k-NN with the k equals to 5. Standing on this point, the improvements which Novelty detection brings us are useless in these experiments.

Then we go back to discuss the results from enhances k-NN classifier with k equals to 5 and 10. From the Table 4-5, we find the accuracy of ham classification reaches to 99.8% which is the best results from all classifiers we implemented in Chapter 4. However, the excellent performance of ham classification trades of the bad performance of spam classification where the accuracy of spam classification is beneath 10%.

However, our work isn't in vain. As we described in the previous section, the ham classification is as important as spam classification due to we can't lose important information from this kind of classifications. So if we could do something with the spam posts to make it more distinguishable from ham posts, the k-NN classifier with Novelty detection approach can be a quite good alternative to Naïve Bayes classifier.

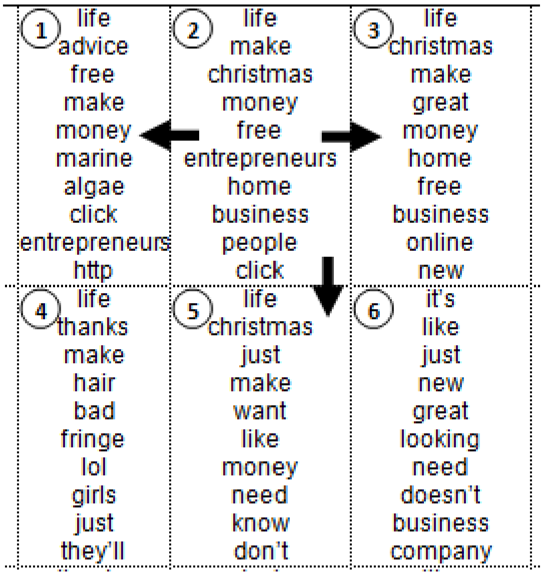
5.4 SOM classifier

Usually, researchers making use of SOM to map high-dimensional data to low-dimensional grid and visualize the low-dimensional grid intend to group vast amount of data into clusters. In this project, we try to make use of the results of SOM clustering to classify spam and ham posts. The following sections will discuss our work in detail.

5.4.1 Map typical terms

In this project, in each node of SOM, there are several terms which are considered as the characteristic terms of this node. In Chapter 4, we fully described the method to obtain these terms and now we may deeply discuss the distribution and usage of these terms.

From Figure 5-6, we will see six grids, which are nodes in SOM, in the picture and each of them comes with a sequence number (we labelled them when we produced this picture). From these terms, we strongly feel that terms which include ‘free’ and ‘money’ should be classified as spam terms and which include ‘lol’ should be classified as ham.



1 life advice free make money marine algae click entrepreneurs http	2 life make christmas money free entrepreneurs home business people click	3 life christmas make great money home free business online new
4 life thanks make hair bad fringe lol girls just they'll	5 life christmas just make want like money need know don't	6 it's like just new great looking need doesn't business company

Figure 5-6 SOM node demonstration

But about which terms belongs to ham and which terms belongs to spam, where do we get these feelings? From the native of spamming and target consumers considered by spammers, we may have the following experiential conclusion: spamming about selling products (due to the limited space, we only use spamming behaviour targeting products advertisement as an example) is always using low price or even free service to attract consumers which is to say the writing will end in using terms like ‘free’ or ‘cheap’ with ‘money’ or ‘price’ a lot.

On the contrary, ordinary online discussions include various kinds of topics so we actually couldn’t give any conclusions about the terms people use but we have experiences in another direction to distinguish spam and ham — emoticons. People usually include some emoticons in the writing to express their feelings about the topic or other people like ‘lol’ or ‘:)’ indicating the delighted emotion and ‘:(’ indicating the unpleasant emotion. These emoticons are regularly used by discussion participators but seldom used by spammers.

From the above illustration, we could simply carry out the node classification manually but this will be a lot of work depending on the size of map and terms in each node. After the elaboration of features for nodes in Figure 5-6, we will explain the method we used to carry out the node classification.

Take the first row as an example, from node ‘1’ to node ‘3’, each of them has some common terms with its neighbour. And if we focus on node 2, it has common terms with both sides of its neighbours and it even has common terms with the node beneath it. Back to this row, we could find out that terms are changing slightly from left to the right what we’ve seen in [12] before and this process happens to the second row as well.

From the SOM theory, we know that, each node has the opportunity to be chosen as BMU, so each node in this map may have common terms with its neighbours. In this case, the node 2 has common terms with all its five neighbours.

Since our SOM classifier combined with Naïve Bayes classifier, these nodes are classified by Naive Bayes classifier based on the same training set used in pure Naïve Bayes classifier. And the result is in Table 5-1. Unlike node ‘2’, node ‘4’ and ‘6’ do not contain terms like ‘free’ and ‘money’ which are considered as high value terms in spam posts.

Naïve Bayes	Nodes
Spam	1, 2, 3, 5
Ham	4, 6

Table 5-1 SOM node demonstration

The classification result doesn’t coincidently match our feelings or to say our feelings come from our experiences with online discussions. As we explained in the SOM classifier prototype design part, the classification for each node used to be done manually. However, the number of nodes is large as well as many terms in each node which may cause man made wrong categorizations. In addition, we believe that automatically classify each node will be the right direction in text classification using SOM approach.

5.4.2 Classification accuracy

In Table 4-7, the accuracy of classifying ham is 98% but the accuracy of classifying spam is 4.8%. The first accuracy is better than pure Naïve Bayes classifier but the accuracy of the second one could be treated as a trade off which is not good. If we stop there, we may say that SOM classifier gives quite good results in ham classification but not so sensitive to spam posts. So, we introduce you to have a look at the second result.

Table 4-9 shows the result from SOM classifier combining with Novelty detection. From the above experiences, we could draw a conclusion that integrate Novelty detection with classifier will obvious improve the performance of ham classification which makes classifier reduce wrong classifications but the trade off of this improvement is slightly reduce the accuracy of spam classification. In this experience, the accuracy of ham classification reaches 94.6% and the accuracy of spam classification is 57.1%. Simply from these figures, only the accuracy of ham classification of Naive Bayes combined with Novelty approach higher than 94.6% while the accuracy of spam classification is also higher than 57.1%. That is to say the SOM classifier with Novelty detection may take the second place in this accuracy competition.

As we explained in the previous chapter, since each time we initiated the map with random weights for each node so nodes may end in different categories each time as consequence. In this case, results from Table 4-7 and 4-9 can't be compared to each other since they came from totally different maps. But we still can obtain much information from these two results.

We discovered that SOM classifier generates different results each time we run the program in section 4.4.2. The consequence of this is we can get different maps each time or to say though the size of map is all the same each time but the terms in each node may very likely not be the same. And we may choose one or more of them as the sources of Naïve Bayes classifier to classify posts. Even though we may say the SOM classifier isn't a stable solution for text classification, it's the approach which can visualize characters of spam and ham in maps.

But speaking to the efficiency of SOM classifier, it takes the longest time to classify posts in our experiments. So, if we can speed up this method, it may perform much better than k-NN or even reach Naïve Bayes. And the optimization work of the SOM classifier should be the focus for a continued study of this problem.

Chapter 6
Conclusion & Future work

6.1 Conclusion

In this thesis, we have studied and invested three pattern recognition theories which are Naïve Bayes, k Nearest-Neighbour and Self-organizing map, as well as Novelty detection approach. Based on SMC environment, we implemented each of these theories into prototypes and also tried to combine Novelty detection approach with them respectively. The purpose of implementing these theories is to filter spam SMCs from our social media and we tested our classifier by using messages from online discussion boards which are one kind of SMC from Integrasco A/S.

To challenge our classifiers, we used various domains of discussion posts collected from normal forums and advertisement forums and we also try to classify discussion posts based on their own threads. By investigating those results obtained from each classifier, we may draw the following conclusion.

The best classifier in our experiments is Naïve Bayes classifier integrated with Novelty detection approach. The accuracy of ham classification increases from 90.5% gained from pure Naïve Bayes classifier to 99.1% while the accuracy of spam classification slightly decreased. The decline performance in spam classification can be tolerated since we already gain a very good result from ham classification part. The reason by saying so is that even if there are still misclassified spam posts in the processed thread, we could tag those confusing ones to indicate the uncertainty from classifiers and we may not mistakenly categorize ham posts into spam which means we have not lost any important data. And more importantly, the Novelty detection approach allows us to consider the candidate posts in the context-based environment and carry out the classification process based on the whole thread or to say the context of the candidate post.

Regarding to the pure Naïve Bayes classifier, it also generates quite promising results which is showed above and we also have a very thorough discussion in Chapter 5. Besides the classification results from our classifier, we have other findings. Actually, to improve the performance of pure Naïve Bayes classifier we have two approaches to choose from our project: the first one is to train the classifier with the data in the same domain where the data in the test set is and this approach has been discussed in section 5.2.1; the second approach is to integrate Novelty detection with the pure Naïve Bayes classifier and this one is the one we chose to use in this project which we have already discussed elaborately.

Speaking to k-NN classifier, it gives good result when we select k to be 5 and the result is even competitive to Naïve Bayes. But, it also takes much longer than Naïve Bayes to get this result. With Novelty detection approach's help, the results are much useful when we set k to 50 or 100. But, compare to the first situation when k-NN works without Novelty detection and get even better results, we decided to say that Novelty detection approach may not be the best consideration when combining with k-NN.

The SOM classifier generates different results each time we run it, so it is not a stable classifier for online discussions. But the basic function which is to cluster vast amount of high-dimensional data is quite helpful when we use it to find the features for each category. Unfortunately, to establish this map may take quite a long time comparing to Naïve Bayes and the classification progress which is similar to k-NN classifier is also a time consuming process. Based on these two points, this approach may not currently be considered as the spam classifier.

Novelty detection approach in this project has been tested with all three theories that prove that this approach is quite helpful when we integrate it with Naïve Bayes classifier and it performs quite well on ham classification based on the context where the candidate post stays. It also shows a great potential when we combine it with k-NN classifier and SOM classifier. And if we could find a better way to weight post in which the spam could be more distinguishable from other ordinary posts, the performance of classifiers integrating Novelty detection would be more attractive.

6.2 Future work

The methods presented in this thesis have demonstrated its performances to classify spam and ham posts from SMCs with some of them having high accuracy and efficiency. Even though we have produced good solutions with our experiments, there is still room for improvements. Here we propose some possible means to optimize our approaches.

Consider using n-gram as feature extractor

With the help of Novelty detection, Naïve Bayes performs quite well in this project. But we still need to improve the classifier to make it more sensitive to spam texts. During our experience, we only use words as the feature extractor, so following work may consider using n-gram as feature extractor when we train the classifier.

Consider better parameter k for k-NN classifier and reducing the time consumption

The k is the most crucial parameter using k-NN classifier to do classifications. In the future work, one may try to find a better parameter which can generate even better results. And to shorten the time used to calculate distance between documents can highly improve the performance of k-NN classifier. Since we have only tried Euclidean distance one may try to use Manhattan distance to measure the distance between documents, etc.

Consider combining SOM with k-NN

During the project, we found that maybe we could make use of the map having terms with each node generated by SOM approach. Since each node has a list of terms which carry the feature of a certain class, we could make use of this list of terms combining with k-NN classifier. Generally speaking, now we calculate the distance between each candidate post and node in the map, but if we calculate each candidate post to all nodes in the map and choose k neighbours to decide the category, we may obtain a better result.

References

[1] Xavier Carreras, Lluís Màrquez, Boosting trees for clause splitting, Proceedings of the 2001 workshop on Computational Natural Language Learning, p.1-3, July 06-07, 2001, Toulouse, France.

[2] Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C., and Stamatopoulos, P. 2000c. Learning to filter spam e-mail: A comparison of a naive Bayesian and a memory-based approach. In Proceedings of the Workshop on Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2000) (Lyon, France), H. Zaragoza, P. Gallinari, and M. Rajman, Eds. 1--13.

[3] About planet.socialmediaresearch.org. [Online]. Available: http://planet.socialmediaresearch.org.

[4] Niu, Y., Wang, Y. M., Chen, H., Ma, M., and Hsu, F. A Quantitative Study of Forum Spamming Using Context-based Analysis. In Proc. Network and Distributed System Security (NDSS) Symposium, February 2007.

[5] Richard O. Duda, Peter E. Hart, David G. Stork, Pattern Classification (2nd Edition), Wiley-Interscience, 2000.

[6] S. B. Kotsiantis, Supervised machine learning: A review of classification techniques, Overview paper, July 2007.

[7] E. M. Voorhees. Overview of the TREC 2002 question answering track. In Proceedings of the Eleventh Text REtrieval Conference (TREC 2002)2002.

[8] Ian Soboroff and Donna Harman, Novelty detection: the TREC experience, Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing, p.105-112, October 06-08, 2005, Vancouver, British Columbia, Canada.

[9] Fabrizio Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys (CSUR), v.34 n.1, p.1-47, March 2002.

[10] Sedding J. and Kazakov D., WordNet-based text document clustering. In Proc. of the 3rd Workshop on Robust Methods in Analysis of Natural Language Processing Data. 2004, 104--113

[11] D. Etzold. Improving spam filtering by combining Naïve Bayes with simple k-nearest neighbour searches. ArXiv Computer Science e-prints, 2003.

[12] T Kohonen, The self-organizing map, *Proceedings of the IEEE*, 1990.

[13] MERKL, D. 1998. Text classification with self-organizing maps: Some lessons learned. *Neurocomputing 21, 1/3, 61-77*.

[14] Giridhar Kumaran and James Allan, Text classification and named entities for new event detection, *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, July 25-29, 2004, Sheffield, United Kingdom.