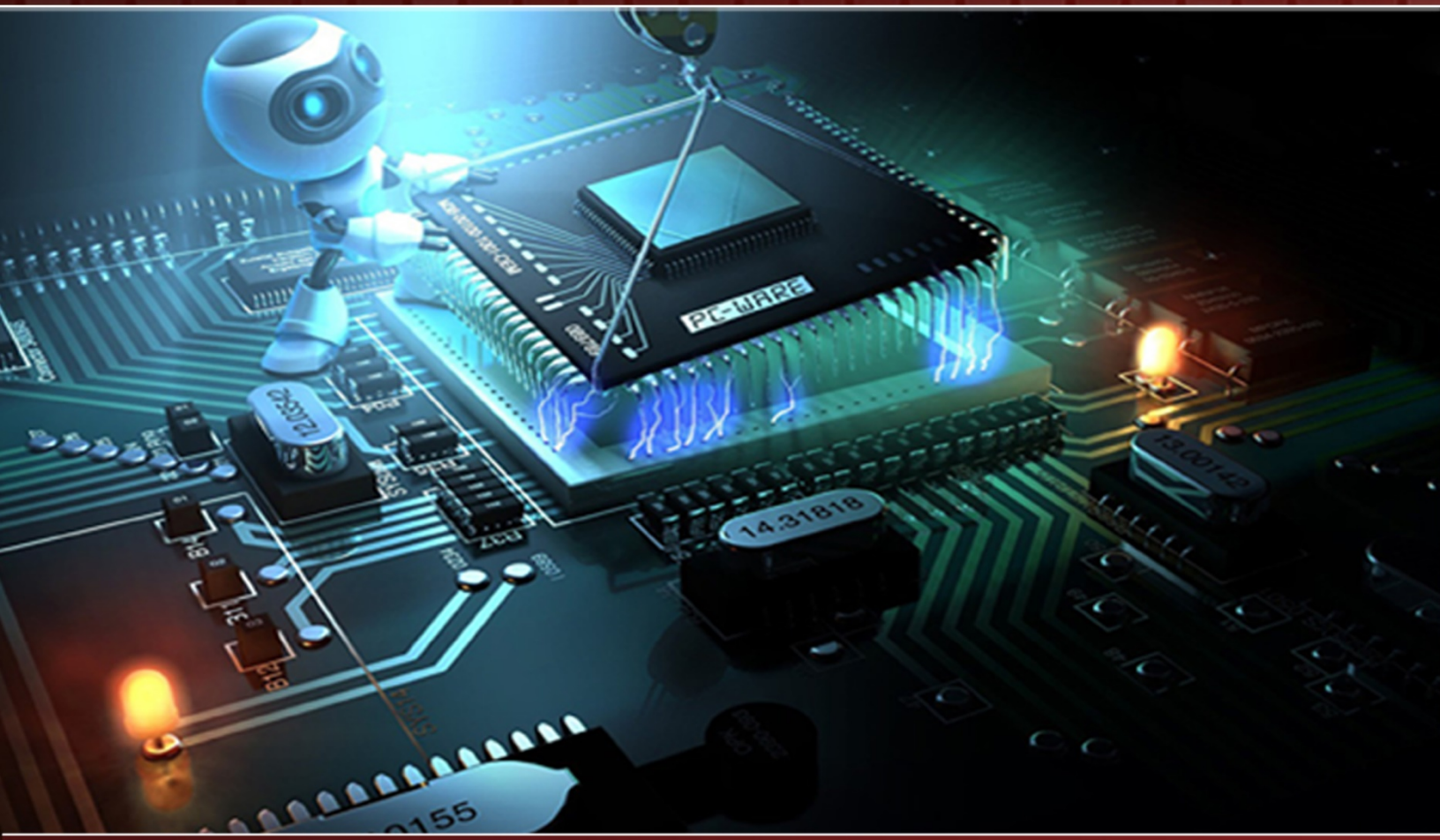
ISSN: 2054-7390

Volume 2. Issue 4 - August 2014

Transactions on Machine Learning and Artificial Intelligence



Society for Science and Education
UNITED KINGDOM

TABLE OF CONTENTS

EDITORIAL ADVISORY BOARD

DISCLAIMER

All the contributions are published in good faith and intentions to promote and encourage research activities around the globe. The contributions are property of their respective authors/owners and the journal is not responsible for any content that hurts someone's views or feelings etc.

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 2, ISSUE 4
ISSN: 2054 - 7390

A Critical Review of the Unguided Loose Search (ULS) Process for Natural Language Based Extraction Technique on Relational Databases

[1]Enikuomehin A.O., [2]Sadiku A.S., [3]Egbudin M.D
[1]Department of Computer Science, Lagos State University, Lagos Nigeria
[2]Department of Computer Science, University of Ilorin, Ilorin, Nigeria
[3]Infinity IT, Lagos Nigeria , Nigeria
toyinenikuomehin@gmail.com, assadiku@unilorin.edu.ng, dechyzeay@yahoo.com

ABSTRACT

Formulation of query statements by searchers for submission into relational databases and information retrieval systems have been a serious challenge that often lead to irrelevant search results. This is compounded by the level of uncertainty about the user's information need and in some cases, unfamiliarity with retrieval system. Evidently, the World Wide Web presents a more established challenge in this area, considering the fact that searchers has little or no training on search techniques on the web. This paper recognizes fuzzy logic system and fuzziness as a tool required to close the gap between automated systems and human thinking. We realize this stiffness in query presentation as against the flexibility in human thinking and then consider the fuzzy concept as a tool that can be incorporated into a new system to overcome the syntactic problem presented in most relational operations. Thus, the paper proposes a novel approach of natural language query based problems. We propose the use of an Unguided Loose Search (ULS) which involves the use of local appropriator on a fuzzified Natural Language Interface. Our approach incorporates fuzziness in the interface, using the local appropriator, of the database systems rather than within the data itself. It allows freedom to users since they will not have to learn any specialized syntax such as that of SQL. The result shows that the new querying model called the EFUSQL model is applicable to real life users and can be incorporated into existing databases and query interfaces. The results show that naïve users prefer the new system due to its flexibility and response time.

1 INTRODUCTION

Human are vague in nature. Incomplete, Imprecise and uncertainty that result from users intention (vagueness) for querying databases has largely not be handled with the existing querying system since queries are not discriminated. There is an extensive research examining

DOI: 10.14738/tmlai.24.308
Publication Date: 3rd Aug 2014
URL: http://dx.doi.org/10.14738/tmlai.24.308

how imprecise and uncertain data can be presented in, and queried from databases given that it is pervasive in most real-world applications. Data extraction, document retrieval, query execution and answering etc have been extensively studied. It has mostly involved the transformation of an information need to a syntactical form in terms of query, in this the user need to have a prior knowledge of the database domain. If we consider as example, the search for the age of a student in a particular university, we will need to formulate a query that will have a target table on a particular database. Problem of execution exist when the user does not know the database table name. It means that the degree to which the query represents a user's information need is a function of user's ability to precisely formulate the information need in a suitable syntactical form admissible within the sql syntax formulation domain. Similarly, many research efforts have been carried out to extend the database models and query languages in other to incorporate fuzzy representation and query capabilities for fuzzy data, the argument still remains that, it is challenging to present a system that is useful and practicable to users since research has been on querying fuzzy databases with some level of fuzziness whereas most real databases are not fuzzy and database administrators still prefer the use of crisp database as evident in most organizations. Furthermore, while some users may want to continue using the traditional facilities available in most of the key commercial database systems and not interested in learning new query languages, other groups of users may want to use fuzzy linguistic terms for querying a non-fuzzy database. This is the case with sql and its extentions. First, users need to know about the existence of the data and secondly, they need to be equipped with technicalities to retrieve such data. This work concentrates on the latter task. Today, one of the biggest challenges in web technologies is data retrieval in multidimensional databases. A multidimensional database is a database that hold data as text in document, Images, video etc, it is a relational database with external links with other databases. To cope with this information growth, existing search methods will need to be enhanced to appreciate an acceptable level of relevance. Query results do not satisfy the users to a large extent thus users are forced to make a decision or a choice based on the displayed output. The same problem occurs also within an organization when a staff is searching for a data from a single database. This can best be explained by the incident of the December 25th 2009 bombing attempt of a Detriot, USA based flight by a Nigerian. In that incident, which can attributed to Query misrepresentation, Adam Brookes released a bulletin on the said bombing of the flight 253 "Once again, it is the failures of the US intelligence agencies that, we are told, are to blame. The report found out that the US government did have 'sufficient information' to disrupt the Christmas day attack. But that information was scattered around databases. It was never pulled together to present a coherent picture of the threat. A 'series of human errors' occurred, apparently someone misspelled Umar Farouk Abdulmutallab's name as they entered it in a database and that is why no-one realize he had a US visa. The above is a clear indication of the problems this paper attempts to resolve. These could exist in a singular database or also a multidatabase system evenly spread and distributed on the internet. Therefore , the major

context here is not the web usage but the perfect method and tools for extraction of the open and hidden data. In most cases, these data are available, however the tools such as the query language, need to be sufficiently enhanced to be able to provide a human usable result. This clearly exceed the capability of the existing database querying tool ' THE SQL and its extentions' . Web interfaces to databases are relatively simple and restricted. Even a skilled user could not define complicated queries due to their limitations. therefore, databases that need to be accessed via the Web should offer more "intelligence". In this paper, we are shifting the intelligence from the database to the query since most relational database users still use the conventional databases and not much of fuzzy databases or any other form of databases that has so much be covered in literature.

2 FUZZINESS IN DATABASES

2.1 Theory of Fuzzification

The original interpretation of Fuzzy sets arises from a generalization of the classic concept of a subset extended to embrace the description of "vague" and "imprecise" notions. This generalization is made in he following way:

1. The membership of an element to a set become a "Fuzzy" or "vague" concept. In the case of some element , the issue of whether they belong to a set may not be clear.
2. The membership of an element may be measured by a degree, commonly known as the "memebership degree" of that element to the set, and it takes a value in the interval [0,1] by agreement.

Using classic logic, it is possible to deal only with information that is totally true or totally false; it is not possible to handle information inherent to a problem that is imprecise or incomplete, but this type of information contains data, which would allow a better soution to the problem. In classic logic the memebership of an element to a set is represented by 0 if it does not belong and 1 if it does, having he set {0,1}. On the other hand, in Fuzzy logic this set extends to the interval [0,1]. Therefore, it could be said that Fuzzy logic is an extension of the classic systems. Fuzzy logic is the logic behind approximate reasoning instead of exact reasoning. Its importance lies in the fact that many types of human reasoning, particularly the reasoning based on common sense are by nature approximate.

Note the great potential that the use of membership degrees represents by allowing something qualitative (Fuzzy) to be expressed quatitatively by means of the membership degree. A Fuzzy set can be defined more formally as follows:

Defination 1:

A **Fuzzy set** A over a universe of discourse X (a finite or infinite interval within which the Fuzzy set can take a value) is a set of pairs

$$A=\{\mu_A(x) \,/\, x : x\in X, \mu_A(x)\in[0,1]\in\Re$$

OR (1)

$$\mu_A(x)=\begin{cases}1, iff\ x\in A\\ 0, iff\ x\in A\end{cases}$$

Where $\mu_A(x)$ is called the memebership degree of the element x to Fuzzy set A. This degree ranges between the extremes **0** and **1** of the dominion of the real numbers:

- $\mu_A(x)=0$ indicate that x in no way belong to the Fuzzy set A
- $\mu_A(x)=1$ indicates that xd completely belongs to the Fuzzy set A

Sometimes, instead of giving an exhausive list of all the pairs that make up the set (discreet values), a definition is given for the function $\mu_A(x)$, referring to it as characteristic function or **membership function.**

2.2 Fuzzy sets Theory and Query processing in Databases:

In written sources, we can find a large number of papers dealing with this theory, which was first introduced by Lofit A. Zadeh in 1965 (1). A more modern synthesis of fuzzy sets and their applications can be found in (2), (3), (4) , (5), (6) .

The original interpretation of fuzzy sets arises from a generalization of the classic concepts of a subset extended to embrace the description of “vague” and “imprecise” notions. This generalization is made considering that the membership of an element to a set becomes a “fuzzy” or “vague” concept. In the case of some elements, it may not be clear if they belong to a set or not. Then, their “membership degree” of the element to the set, and it takes a value in the interval [0,1] by agreement. Using classic logic, it is only possible to deal with information that is totally true or totally false; it is not possible to handle information inherent to a problem that is imprecise or incomplete, but this type of information contains data that would allow a better solution to the problem.

Querying is the process of retrieving information or data from the database. The traditional query in a relational database has been shown to be incapable to satisfy the needs for dealing linguistic values. The structured query language Sql has been around for while from RDBs, the Sql is a declarative language that allows the user to specify “what” information from the database is needed without having to specify how it is to be retrieved: IE is constructed such that each DBMS translate individual query into an efficient execution plan. Recall that these were issue of bi-valued interest if an item belongs into a set or not since the time of Aristotle’s there. The answer has been usually formed as a simple truth function assuming only values YES or NO for an answer. A thought shift has been made in 30’s of the last century particularly by Lakeview 2’s thesis. The contemporary SQL norm supports only classical bi-valued logic. Unfortunately, the use of fuzzy sets and fuzzy logic operations is not defined and there are

many of mutually different commercial and General Public License SQL server distribution in essential SQL norm implementation.

3 APPROACH

3.1 Retrieval based on Similarity

The approach used is an extension of the concept of retrieval based on similarity as a function of relevance. This is used in the formulation of an appropriate model as a benchmark for the work. Relevance is measured by concept of similarity. There are two type of relevance in similarity:

3.1.1 Fuzzy similarity

Definition 2 .

Let L be a Fuzzy algebriac function and let A be a non-void set. A fuzzy similarity S on A is such a binary fuzzy relation that, for each x, y, and z in A,

i. $S(x,x) = 1$ (everything is similar to itself),

ii. $S(x,y) = S(y,x)$ (fuzzy similarity is symmetric),

iii. $S(x,y) \circ S(y,z) \leq S(x,z)$ (fuzzy similarity is weakly transitive)

3.1.2 Data Similarity

For every value 't' in the domain of attribute 'A', D(t) can be defined as $\log(n/F(t))$,

where 'n' = number of tuples in the database

F(t) = frequency of tuples in database where 'A' = 't'

The similarity between a tuple 'T' and a query 'Q' is defined as: i.e., similarity between a tuple T and a query Q is simply the sum of corresponding similarity coefficients over all attributes in T.

3.2 Computing top-k Answers

Assume a query Q with m elementary conditions on the attributes A_i, i in {1,…,m}. The multidimensional database D consists of a single relation R with a finite set of N tuples described on the attributes $A_1,\ldots,A_m$. Each tuple t is associated with a vector $(x_1,\ldots,x_m)$ of m scores, one for each attribute of the elementary query condition. Scores are computed from attribute values of each tuple with respect to their similarity to the query condition. For the top-k problem, the database could alternatively be seen as a set of m sorted lists L_i of N pairs (t, x_i), t in R. Hence, for each elementary condition of the query Q, there is a sorted list L_i in which all N database tuples are ranked in descendant order. Entries in the lists could be accessed randomly from the tuple identifier or sequentially from the sorted score. The main issue for

top-k query processing is then to obtain the k tuples with the highest overall scores computed according to a given aggregation function $agg(x_1,\ldots,x_m)$ of the attribute-oriented scores x_i. The aggregation function agg() used to combine elementary conditions has to be monotone; that is, agg() must satisfy the following property:

$agg(x_1, \ldots, x_m) <= agg(x_1',\ldots, x_m')$ if $x_i <= x_i'$ for every i. (2)

Among the various monotone aggregation functions are t-norms and t-conorms respectively associated with conjunctive and disjunctive queries, and weighted means as well. Min and Max are the most common functions respectively for conjunctive and disjunctive queries. The naive algorithm consists in looking at every entry (t, x_i) in each of the sorted lists L_i, computing the overall grade of every object t, and returning the top answers. Obviously, this approach suffers from a high access cost to the lists since all the N overall grades are computed.

Let us address set of all N tuples in a relational database table as the tuple table, sorted by the decreasing order of score. We store information about the x-tuples in an x-table. By using a hash map, given the id of a tuple t, the score and confidence values for all its alternatives can be retrieved efficiently from the x-table in O(1) time(Worst Case).

To process a top-k query, we retrieve tuples in decreasing score order and stop as soon as we are certain that none of the unseen tuples may possibly affect the query result. The dept of the search denoted by n can be defined as the minimum number of tuples retrieved so as to generate the correctness of the result. The k top approach can be used to fomalize a model presented below:

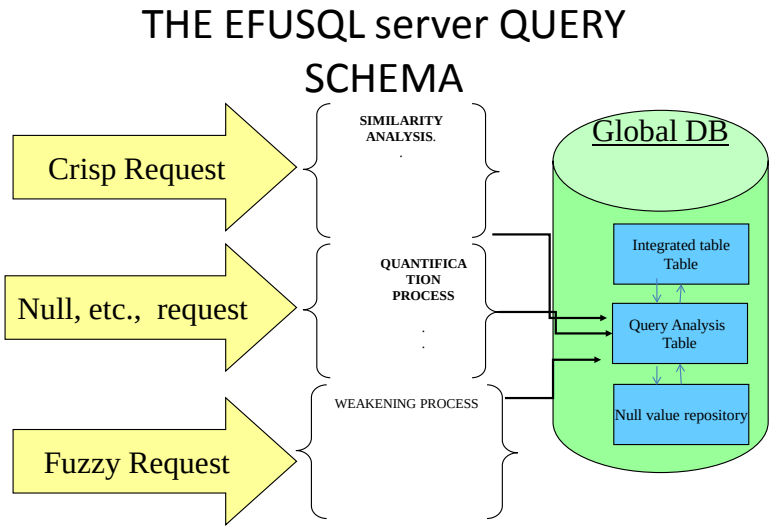


Figure 1: A similarity system on a multidimensional DB

4 EXPERIMENT

An application interface developed in php is placed over the sq1 server for the purpose of translation. We did not use the Fsq1 server of medina as it has not been fulfilled practically and more so 77% of RDBMS users still use the conventional sq1 server and no Fsq1 server. It acts as a middle ware that translates the National Language to crisp sq1. Consider the query:

Brilliants students that are young:

Pre query analysis:

"Student" is the key word; therefore it is awarded a relevance 'Mark R'.

The system analyses the query by gathering information from different data sources, combine them to a standard format then store on a location. The associated sample table is shown in table 1.

Table 1: Database table to model performance

NAME	CGPA	LEVEL	STATE OF ORIGIN
Toyin	2.70	300	Ondo
Usman	3.40	400	Kano
Smith	2.50	300	Lagos
Kunle	4.00	200	Kwara
Shade	2.67	100	Oyo
Nnena	3.35	200	Imo
Obiora	2.21	200	Anambra

In the process of establishing a truly fuzzy querying system, other information than the crisp inserted data may be required: As example, for state of origin, one way need local government area, or village name or family house. This other information are called Meta information since they provide more information about the data. The introduction of metadata in commercial search engine is a novel idea whose popularity has not been fully harnessed. Its importance is in its ability to provide non discrete information about data. The meta table contains all the information required for fuzzification in stage (3). The meta-table is given as follows:

Linguistic –Term: used to store the name of the fuzzy set.

Table- name: used to refer name of the table in which attribute associated with the fuzzy set is available.

Column-name: used to refer to attribute associated with fuzzy set.

Alpha (α): lower range of the SUPPORT [Ross, Fuzzy logic in Engineering]

Beta (β): lower range of the CORE [Ross, Fuzzy logic in Engineering]

Delta (δ): upper range of the CORE [Ross, Fuzzy logic in Engineering]

Gamma(γ): upper range of the SUPPORT [Ross, Fuzzy logic in Engineering]

The core of a membership function for some fuzzy set A is defined as that region of the universe that is categorized by a complete and full membership in the set A. that is the CORE comprises those elements x of the universe such as that $\phi_A(x) = 1$.

SUPPORT the region that is characterised by non zero membership in the set A i.e. $\phi_A(x) > 0$.

The core of a membership function for some fuzzy set A is defined as that region of the universe that is categorized by a complete and full membership in the set A. This approach is different from the earlier approaches as in (7), (8) where membership functions wer not used, however thresholds were included. At the end of it, after membership fixes are generated, the values are now defuzzified and ranked in ascending order.

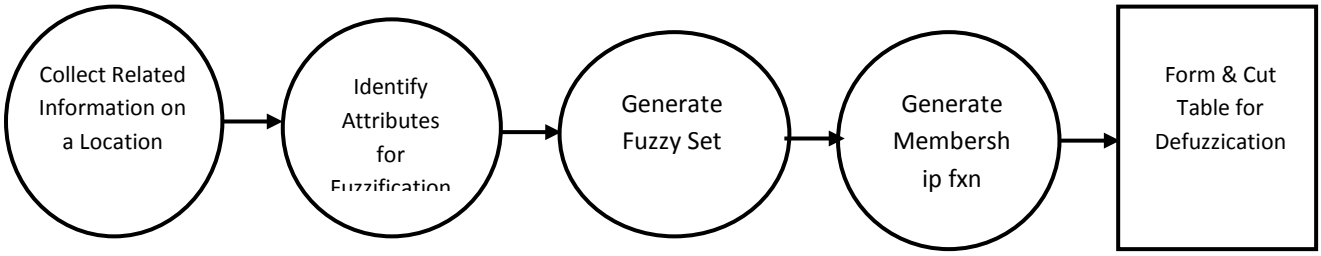


Figure 2: The data data fuzzification flow

5 COMPUTATION OF MEMBERSHIP FUNCTIONS

To allocate membership value, the fuzzy sets generated is divided into three groups;

Group A:

The set of terms and values at the lower boundary region.

Group B:

Th The set of terms and values at the is forms the core region

Group C:

The set of terms and values at the upper boundary region.

The membership value for the core region is always to 1. The lower boundary and upper boundary are calculated as

$$(x- \alpha)/(\beta - \alpha) \; and \; (\delta - x)/(\delta - \gamma)$$

where x is the value of the attribute that can be brought from the concerned table.

To incorporate weight, we consider a fuzzy relation R such as shown in table 2 below and called the grading factor used in generating the values in table 1:

Table 2: Grading Factors

		Excellent	Very Good	Good	Fair	Poor
	Correct answer	0.3	0.4	0.3	0.1	0
R =	Language	0.0	0.2	0.5	0.3	0
	Presentation	0.1	0.6	0.3	0	0

Now, the professor want to assign a grade to each paper, we formalize this approach, thus Let X be a universe of factors and Y be a universe of evaluations, so

$$X = \{ x_1, x_2, \ldots\ldots\ldots x_n\} \text{ and } Y = \{y_1, y_2, \ldots\ldots\ldots\ldots\ldots y_n\} \quad (4)$$

where i = 1, 2, … n and i= 1, 2, … m.

Suppose we introduce a specific paper into the evaluation process in which the professor has given a set of "scores" (w_i) for each of the n grading factors, we ensure, for convertion, that the sum of the scores is unity. This means that each of the scores is actually a membership value for each of the factors xi, and they can be arranged in a fuzzy vector $\underline{\omega}$, to have,

$$\underline{\omega} = \{w_1, w_2, \ldots\ldots\ldots, w_n\} \quad (5)$$

where

$$\sum_i w_i = 1$$

The process of determining grade for a specific paper is equivalent to the process of determining a membership value for the paper in each of the evaluation categories, y_i. This process is implemented through the composition operation

$$\underline{e} = \underline{\omega} \text{ o } \underline{R}$$

where e is a fuzzy vector containing the membership values for the paper in each of the y_i evaluation categories.

After the generation of the membership function and the weight, we compute the membership values corresponding to the attribute. To do this, we propose the system checks for hedges (fundamental atomic term are often modified with adjectives (nouns) or adverbs (verb) like very low, slight, more or less, fairly, almost etc, that is, the singular meaning of an atomic term is modified or hedged) from its original interpretation. Using fuzzy set as the calculus of interpretation those linguistic hedge have the effect to modify the mf for a basic atomic term (9). When the hedges is calculated and manipulated, the next stage removes the fuzziness contained in the query by the use of appropriate deffuzzication technique. However, in our model, we introduce a Hard Aggregation Concepts (HAC) before the deffuzzication processes take place. In the Hard Aggregation Concepts (HAC), other components of the query are

segmented in parts. These parts are the Modifier Part (MP) and the Concepts Part (CP). We propose

$$MP + CP = HAC \tag{6}$$

In practice, there might be more than one HAC in a query, then we say,

let there exist HAC such that any term in query is an element of the HAC, representing the i=th HAC. To explain our idea of the HAC, let use the following example;

Find the name, level, age of very young and quite tall students where grade ≥ 4.0.

The HAC can be implemented as follows;

[name, age, grade] are return attributes

[very, quite] are Modifier Parts

[Young, tall] are Concept Parts;

Student is a table on the database

Grade ≥ 4.0 is a Crisp Condition.

After this "Classification", the above computation of membership function is carried out on the HAC components and the values are now integrated into the query range. Then deffuzzification is then implemented by finding the α - Cut and by calculating the maximum and minimum range. Once of minimum and maximum range is calculated with the fuzzy terms with linguistic hedges are remodel and the result displayed. This means that the central meaning of the query has been taken into account in the query processing. This work introduces a new process of implementing fuzzy queries with the use of membership value manipulation and HAC. It enables us to write natural language fuzzy queries at frontends and get discriminated results, this also helps in handling missing data since if a row satisfies at least one fuzzy criteria or crisp criteria, it will be accommodated in the range and then will be included in the result set even if the data has some missing attributes.

An important aspects of this research is that this query system can process multiple fuzzy queries in plain in human language as well as crisp query at the same time with optimum intelligent result. Steplan et al (01) presents a Skyline operator for air flight selection, which select best rows or all non-dominated based on a crisp multi-criteria comparism. A row dominates the other if it is as good or better than the other in all multiple criteria and better in at least one criterion.

6 CONCLUSION

In the implementation of the suggested technique, the followings significant observations were made; the query language used is highly flexible i.e. user need not bother about the syntax of the language, queries may contain fuzzy, uncertain and imprecise terms. Database schemas

need not be modified for storing the membership of individual record, Fuzzy extensions are being created automatically so overhead of neither user nor DBA is being increased, Updating of fuzzy extensions after each and every update in master database is also being done automatically with the help of triggers, In the implementation of this technique only logical view of database is being affect(Table referencing), Crisp queries are also being handling along with the fuzzy ones. Intelligence has now been incorporated with searching to get result much more human. The paper, following these conclusions recommends that non crisp processes should be included in the highly patronized commercial database.

REFERENCES

[1]. Zadeh, L.A. "Fuzzy sets". Information and control, 8, 1965, pp. 338-353

[2]. Buckley, J.J., & Eslami, E. "An Introduction to fuzzy logic and fuzzy sets" (advances in soft computing). Physica-Verlang Heidelberg, 2002

[3]. Kruse, R., Gebhardt, J.,& Klawonn, F. "Foundations of fuzzy systems". John Wiley & Sons, 1994

[4]. Mohammad, J., Vadiee, N., & Ross, T.J.(Eds.). "Fuzzy logic and Control: Software and Hardware applications". Eaglewood Cliffs, NJ: Prentice Hall PTR, 1993,

[5]. Nguyen, H.T., & Walker, E.A. "A first course in fuzzy logic (3rd ed.)". Chapman & Hall/CRC, 2005, Piegat, A. "Fuzzy modeling and control". Physica-Verlag (Studies in Fuzziness and Soft Computing), 2001

[6]. Pedrycz, W., & Gomide, F. "An introduction to fuzzy sets: Analysis and design" (A Bradford Book). The MIT Press, 1998].

[7]. Galindo, J., Medina, M., Pons, O., & Cubero, J. (1998). A Server for Fuzzy SQL Queries. In T. Andreasen, H. Christiansen, & H. Larsen (Eds.), Lecture Notes in Artificial Intelligence (Vol. 1495, pp. 164-174). Springer.

[8]. Bosc, P., & Pivert, O. (1994). Fuzzy Queries and Relational Databases. Proceedings of the 1994 ACM Symposium on Applied Computing, (pp. 170-174). Phoenix,AZ.

[9]. Galindo, J., Urrutia, A., & Piattini, M. (2006). Fuzzy Databases: Modeling Design and Implementation. Hershey:PA: IDEA Group.

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 2, ISSUE 4
ISSN: 2054 - 7390

A Vision-Based Assistive Robotic Arm for People with Severe Disabilities

Hiroki Higa, Kei Kurisu, Hideyuki Uehara
Faculty of Engineering, University of the Ryukyus, Japan;
hrhiga@eve.u-ryukyu.ac.jp

ABSTRACT

This paper presents a vision-based assistive robotic arm for people with severe disabilities. This system is composed of a robotic arm, a microcontroller, its controller, and a vision-based unit. The main body of the robotic arm that can be contained in a briefcase is about 5 kg, including two 12-V lead acid rechargeable batteries. This robotic arm is also capable of being mounted on a wheelchair. To obtain position coordinates of an object, image processing technique with a single Web camera was used. Position errors in the order of few millimeters were observed in the experiment. Experimental results of drinking water task with able-bodied subjects showed that they could smoothly carry out the tasks. The present results suggested that the resultant position errors were acceptable for drinking water command.

Keywords: Assistive system, Robotic arm, Image processing, Web camera, People with severe disabilities.

1 INTRODUCTION

This paper deals with an assistive robotic arm system for people with severe disabilities. By the cause of stroke or spinal cord injury, there are many people who have paralyzed extremities and who need someone's help. Some of them have strong-minded to be independent of others and to live their own lives. We also live longer than we used to, in step with the advances in medical technology. Many of caregivers are getting older, and it is demanding for them to take good care of people with disabilities due to their advanced ages. It is of significant importance to support lives of elderly persons as one of common issues in developed countries.

Some robots to assist the disabled have been developed [1, 2]. For example, there are Manus Manipulator [3], Raptor wheelchair robot [4], manipulator mounted on wheelchair [5], Handy 1 [6], and My Spoon [7, 8]. People with disabilities using these systems are able to roughly manipulate an object. It is however difficult for the users with the Manus Manipulator and the wheelchair mounted assistive robots to have something to drink and eat. The Handy 1 and My

DOI: 10.14738/tmlai.24.309
Publication Date: 3rd Aug 2014
URL: http://dx.doi.org/10.14738/tmlai.24.309

Spoon require users to prepare foods to be cut and to use the dedicated tray for meals. In addition, they can hardly dine out with these systems.

In order to explore the optimum solution for the above-mentioned problems, we built a prototype of assistive robotic arm for people with severe disabilities [9, 10], such as stroke, spinal cord injury, and muscular dystrophy (MD). Our final goal is to realize and to provide a low-cost and useful assistive robotic arm system to them. The fundamental concept of our assistive robotic arm system is portability. The robotic arm's main body is small enough to put in a laptop computer's briefcase. The feature enables the user to go out to eat at a restaurant with his/her family, which leads to more desirable improvement of his/her quality of life (QOL). In our previous robotic arm system [10], there were some limitations that its motion and the positions of a plastic bottle, plates, and utensils had to be predetermined, because the system was an open-loop system and did not have any feedback device. We applied a vision-based control method with a single Web camera to the robotic arm system in this paper. To evaluate its performance, position coordinates calculations of a plastic bottle were experimentally carried out. Drinking water tasks with three able-bodied subjects were also performed.

This paper is organized as follows. Section 2 summarizes the proposed assistive robotic arm. Experiments are described in detail in Section 3. Sections 4 and 5 provide experimental results about calculations of position coordinates of the plastic bottle and drinking water tasks, and discussions. The final section concludes the paper.

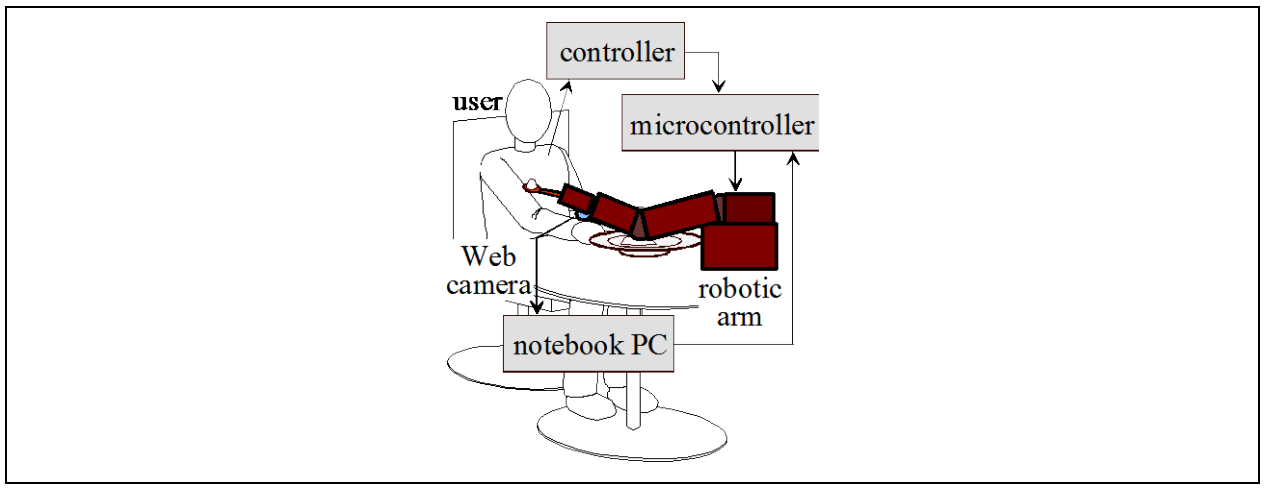


Figure 1: System configuration of assistive robotic arm. The Web camera and notebook PC are used to calculate the position coordinates of an object, which is described in Section 2.2.

2 SYSTEM CONFIGURATION

2.1 System Overview

A system configuration of the proposed assistive robotic arm is illustrated in Figure 1. This system is composed of a robotic arm, a microcontroller, its controller, a Web camera, and a notebook computer. The microcontroller AT91SAM7S256 (Atmel Corporation) has a 32-bit ARM7TDMI RISC processor, which is low-power, small, cost-effective, and good real-time interrupt response. It is embedded to the system.

Figure 2 shows the prototype robotic arm. As can be seen in Figure 2 (a), the robotic arm's main body is totally contained in a laptop briefcase without removing any parts of the robotic arm shown in Figures 2 (b) and (c). The robotic arm also can be mounted on a wheelchair. One of the fundamental concepts for the robotic arm system is portability. This allows user to enjoy not only having dinner with his/her companions in his/her house but also eating out with them. The robotic arm system can be utilized when the user goes on a trip, which enables him/her to try some local dishes at restaurant as much as he/she wishes.

Some of the important technical specifications of the assistive robotic arm are summarized in Table 1. Except for a gripper, the robotic arm has seven servos, which are electromechanical devices in which electrical inputs determine the position of the armature of the motors. These servos are controlled by the microcontroller. The gripper (also called the end effector) is detachable. Its opening and grasping any objects are controlled by a servo (srv 7) which is attached to the wrist portion. The Web camera is fixed to the wrist part of the assistive robotic arm (see Figure 2 (b)). To reduce the weight of the end effector of the robotic arm, some lightweight materials, such as carbon plate, balsa wood, and low foamable vinyl chloride, were used [see Table 1]. We installed an emergency stop switch for emergency purposes. Turning this switch on allows users to immediately shut down the robotic arm system.

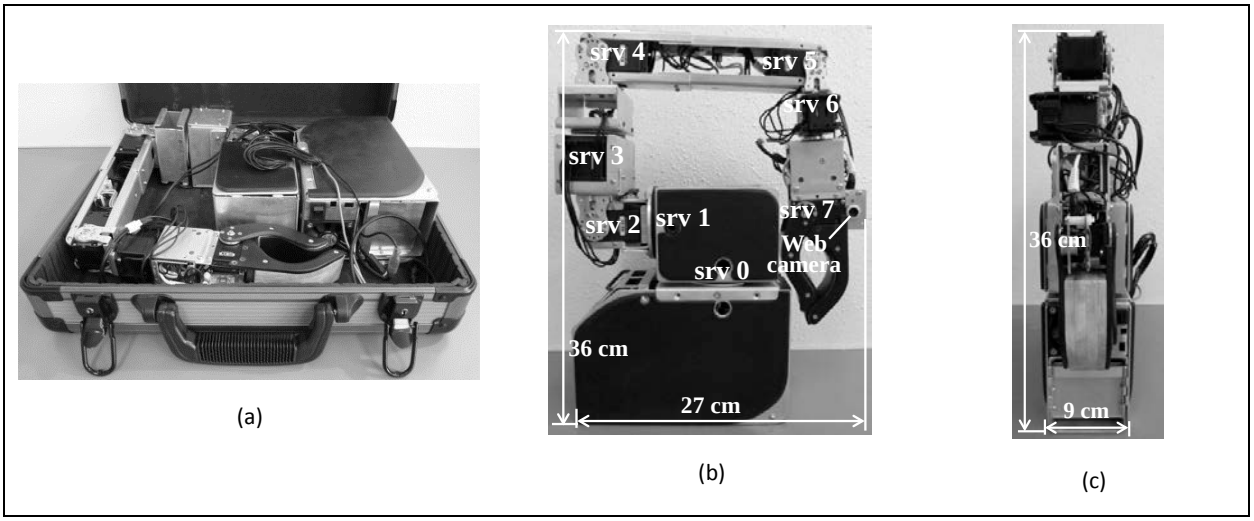


Figure 2: Assistive robotic arm (a) when in a laptop briefcase, (b) its side and (c) rear views. Symbol "srv" in the figure stands for servo.

Table 1: Summary specification of proposed assistive robotic arm

	Specification
Size when folded in a briefcase	36 x 27 x 9 cm
Maximum working area in radius	71 cm
Weight	About 5 kg, including two batteries
Microcontroller	AT91SAM7S256, Atmel
Battery	Two 12-V lead-acid rechargeable batteries
Degree of freedom	7
Web camera	0.3 megapixel
Materials	Aluminum mainly, low foamable vinyl chloride, balsa wood, carbon plate

A control program written in the form of C language was developed with the gcc compiler. The program was then written into a flash memory of the microprocessor which was connected to the notebook PC using a USB cable.

2.2 Calculations of Joint Angles and Plastic Bottle Coordinates

In order to simplify the calculation for controlling the end effector of the robotic arm, its 3-link model in a Cartesian coordinate system in two dimensions is defined in this paper. We have an assumption that the assistive robotic arm is fixed on the user's right side. Figure 3 shows the robotic arm (top-down view) and its simplified 3-link model in two-dimensional Cartesian coordinate system with origin *O* and axis lines *x* and *y*. It is also assumed that the perpendicular line onto the origin *O* (*z* axis shown in Figure 5) coincides with the rotation axis of the servo srv 0, and the angles of servos srv 1, 2, 3, and 6 are fixed during the final stage of reaching an object. The fixation of these angles means that the end effector is moving parallel with the table on which the robotic arm is installed during the final approach to it. l_1, l_2, and l_3 are the link lengths.

A position vector of the tip point P_3 is the center of grasping position of the gripper. When the end effector coordinates x_3 and y_3 are detected, the angles θ_1, θ_2, and θ_3 can be obtained by using inverse kinematics [11]. From Figure 3 (b), the point P_3 coordinates are given by

$$x_3 = x_2 + l_3 \sin\theta_{end} \tag{1}$$

and

$$y_3 = y_2 + l_3 \cos\theta_{end}\,, \tag{2}$$

where θ_{end} is the angle between *y* axis and the link l_3, which is given by

$$\theta_{end} = \theta_{12} + \theta_{23} + \theta_3\,. \tag{3}$$

The angle between y axis and the position vector P_2, θ_{12}, is also expressed as

$$\theta_{12} = \tan^{-1}(x_2 / y_2). \tag{4}$$

By using the law of cosines, the internal angles θ_a, θ_b, and θ_{23} in the triangle OP_1P_2 are given by

$$\theta_a = \theta_{12} - \theta_1 = \cos^{-1}\left(\frac{{l_1}^2 + {l_{O2}}^2 - {l_2}^2}{2 l_1 l_{O2}}\right), \tag{5}$$

$$\theta_b = \pi - \theta_2 = \cos^{-1}\left(\frac{{l_1}^2 + {l_2}^2 - {l_{O2}}^2}{2 l_1 l_2}\right), \tag{6}$$

and

$$\theta_{23} = \pi - (\theta_a + \theta_b), \tag{7}$$

where l_{O2} is the length of the position vector P_2, which is expressed as

$$l_{O2} = \sqrt{{x_2}^2 + {y_2}^2}. \tag{8}$$

We finally have the angles θ_1, θ_2, and θ_3:

$$\theta_1 = \theta_{12} - \theta_a, \tag{9}$$

$$\theta_2 = \pi - \theta_b, \tag{10}$$

and

$$\theta_3 = \theta_{end} - \theta_1 - \theta_2. \tag{11}$$

These calculations are done in the microcontroller. Detecting plastic bottle's mouth coordinates by using the image processing techniques, the joint angles θ_1, θ_2, and θ_3 are determined in this manner.

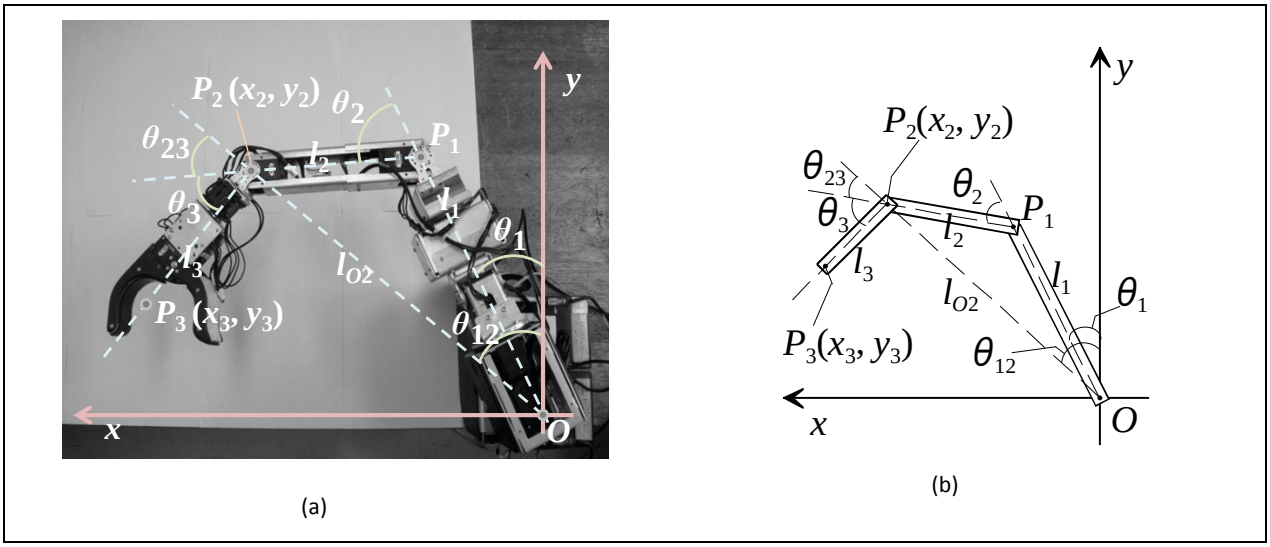


Figure 3: Assistive robotic arm model for the calculation of joint angles in two-dimensional Cartesian coordinate system. (a) Its top-down view and (b) simplified 3-link model.

A flowchart of the image processing to calculate the position coordinates of the plastic bottle is illustrated in Figure 4. First, we calibrate the Web camera set up in the wrist part of the robotic arm, and register the template model obtained from an image of a plastic bottle's mouth. Reading an image, we transform the RGB image into a gray scale image, enhance

contrast of the image, and filter the enhanced image using Median filter [12] in the preprocessing stage. Secondly, the template matching with the registered template model is performed. When a matched area in the preprocessed image is found, calculations of the area and the center coordinates of the plastic bottle's mouth are carried out. We then calculate the height z_h of the plastic bottle using the equation:

$$z_h = 0.211\, A_m + 8.576, \tag{12}$$

where A_m is the area of the plastic bottle's mouth. This calculation is conducted in the notebook PC which is connected to the Web camera. We set a half of the calculated height coordinate as the z-coordinate on the final stage of reaching the plastic bottle.

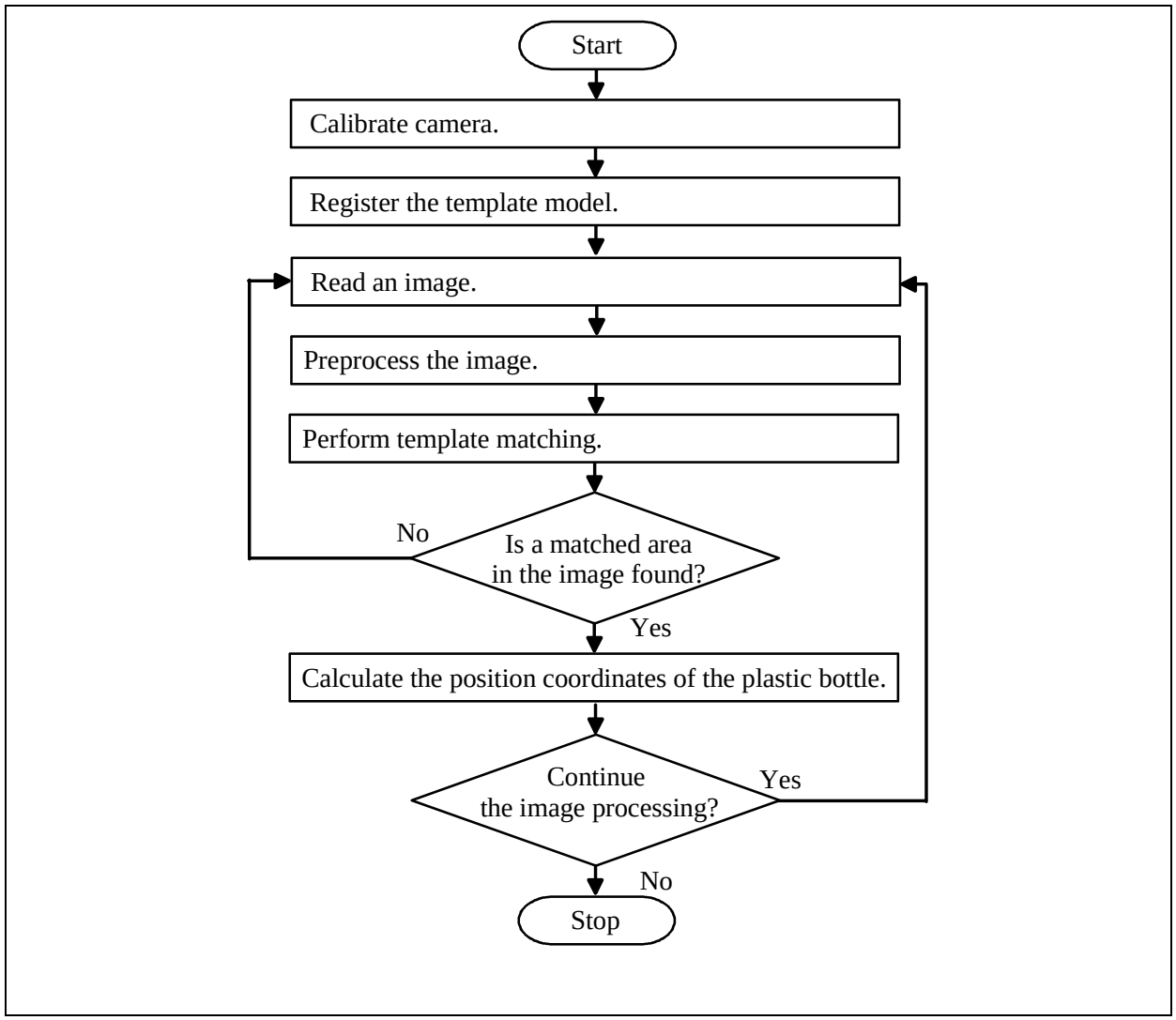


Figure 4: Flowchart of image processing to calculate the position coordinates of the plastic bottle.

3 METHOD

The experimental setup is shown in Figure 5. The robotic arm was placed at the right side. The distance between the camera and the surface on the table is set to approximately 50 cm, capturing an object on the table (Figure 5 (b)). The resolution of images obtained by the Web camera is fixed 352×288 pixels. A white paper is put on the table. Grid lines spaced at 5 cm intervals vertically and horizontally are drawn on it. The grid size is 15 cm long and 10 cm wide. A 500-ml plastic bottle with a black square-shaped coaster was put on 12 intersection points of the two grid lines, and using the image processing technique the mouth of the plastic bottle at each point was detected. The position coordinates of the plastic bottle were then calculated as shown in Figure 4. The height of the plastic bottle is 215 mm. The thickness of the coaster which is placed under it is 3 mm.

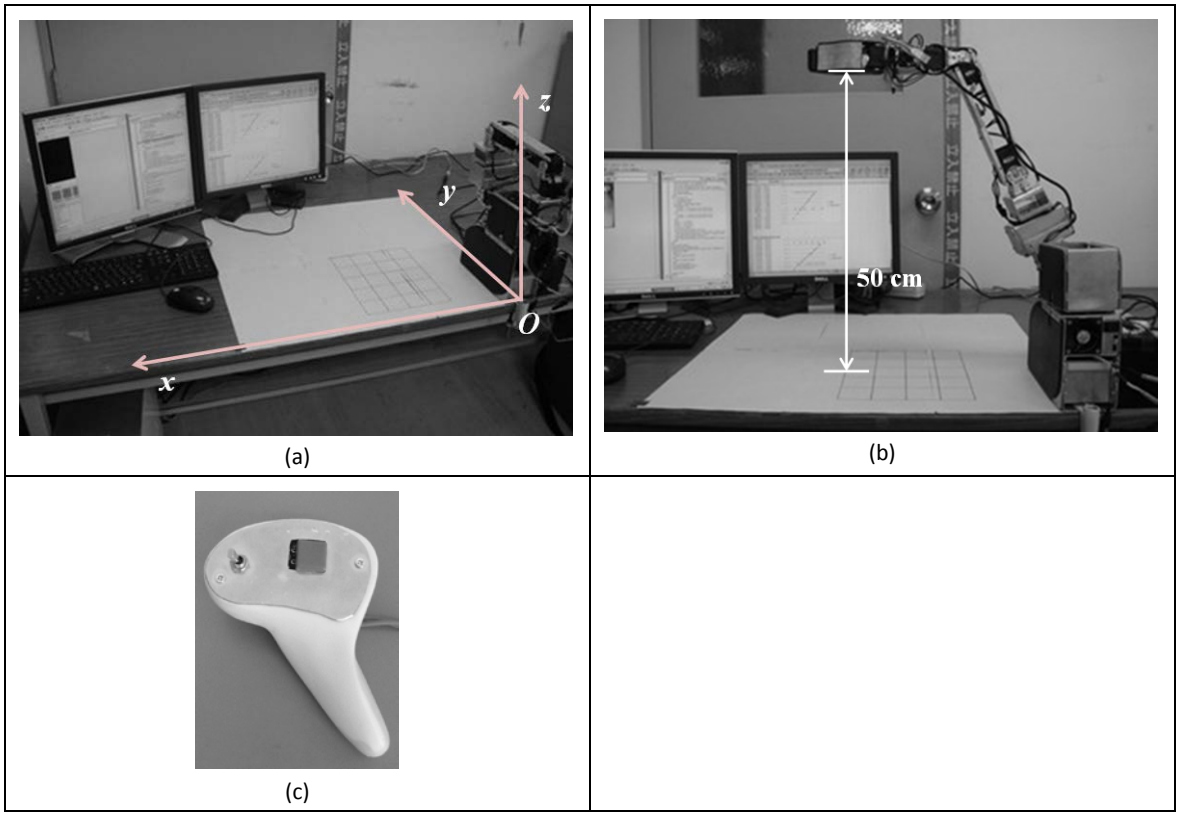


Figure 5: (a) Experimental setup and three-dimensional Cartesian coordinate system, with origin *O* and axis lines *x*, *y*, and *z*. (b) Posture of the robotic arm when an object on the table is capture. (c) Controller used in the experiments.

A drinking water task was carried out next. Experimental setup is the same as shown in Figure 5. Three able-bodied subjects (two males and a female) participated in the experiments. Subjects' age ranged from 22 to 45 years. We obtained informed consents from them. The robotic arm was also placed at subject's right side. Each subject was seated on a chair in front of the table. To issue a command to the robotic arm in the experiments, we made a controller

for the subject as shown in Figure 5 (c). The left toggle switch is the emergency stop switch as described in Section 2.1. The right button is a push button for providing the assistive robotic arm with control commands. The experimental subjects were asked to drink water with the assistive robotic arm. The water was in a 500-ml plastic bottle which was commercially available. The experiments were recorded by a digital video camera.

4 RESULTS

Figure 6 shows examples of the calculated position coordinates of the plastic bottle. The position coordinates on the table where the plastic bottle was placed are indicated in the figure caption, while those of the plastic bottle's mouth obtained by the calculation are shown in the processed images. It was able to detect the plastic bottle's mouth from the images obtained by the Web camera. It could be seen that the position coordinates calculations of the plastic bottle's mouth were properly performed using our image processing method.

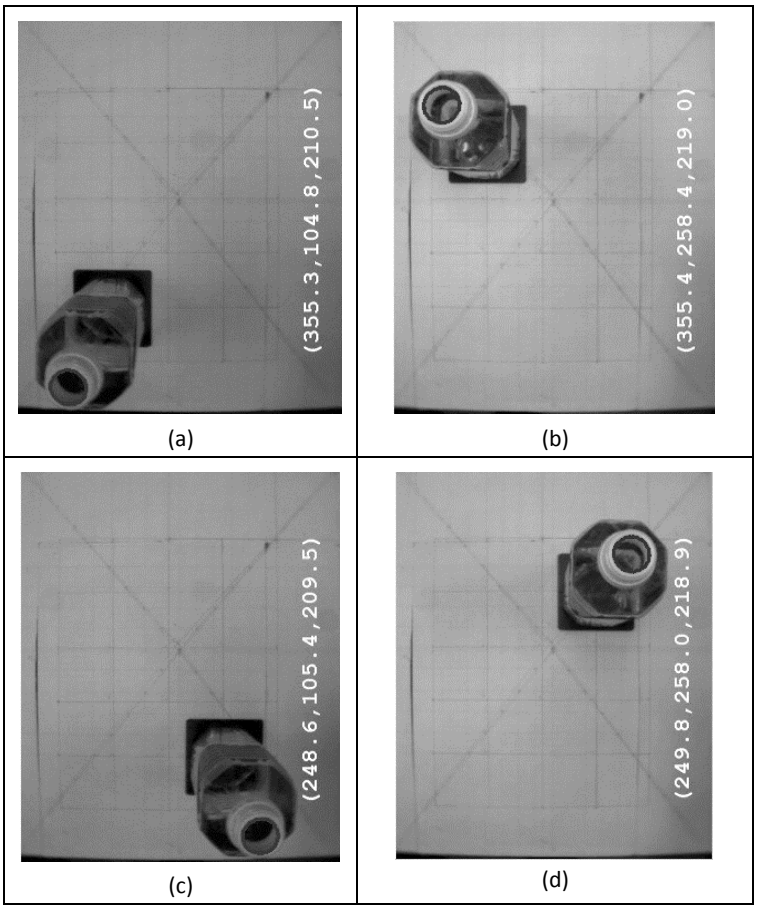


Figure 6: Example results of the processed image with the calculated position coordinates of the plastic bottle's mouth. Its coordinates on the table where it was placed are (a) (350, 100, 0), (b) (350, 250, 0), (c) (250, 100, 0), and (d) (250, 250, 0), respectively. Numerical values in each image mean the calculated position coordinates where the plastic bottle's mouth was. The height of the plastic bottle with the coaster is 218 mm.

Figure 7 represents example result of drinking water task captured from the video data. The elapsed time on each image is indicated in the figure caption. The subject A is drinking water in 19 s (see Figure 7 (d)). The other subjects could also manipulate the assistive robotic arm without any difficulties. It was clear from the experimental results that the subjects could appropriately drink water using the assistive robotic arm system.

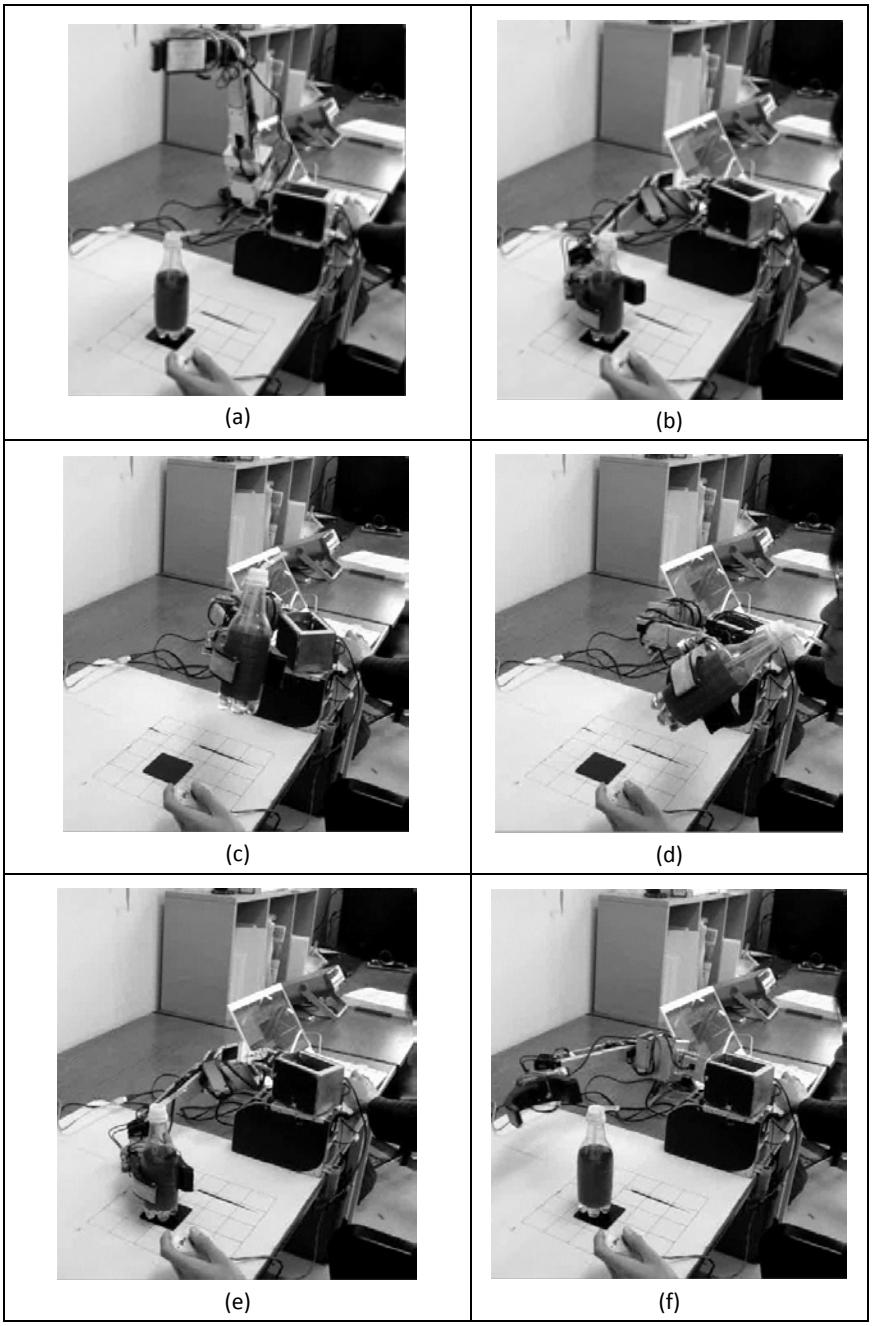


Figure 7: Example of drinking water task performed by the subject A. The elapsed times of the experiment are (a) 0 s, (b) 7 s, (c) 15 s, (d) 19 s, (e) 33 s, (f) 37 s, respectively. These are captured from the video data.

5 DISCUSSIONS

Table 2 lists the averaged errors (means and standard deviations; SDs) in position coordinates on each axis. It was seen from the processed images (Figure 6 and Table 2) that the plastic bottle's mouth at each point was appropriately detected, and that its position coordinates with the errors of few millimeters were obtained. The experimental results of drinking water tasks showed that the able-bodied subjects could grasp the plastic bottle and drink water from it by manipulating the vision-based assistive robotic arm. The time to drink water can be determined by each subject. Hence, it was found from the experimental results that our system allows users to drink water at their paces. In general, the number of drinking water a day is more often than that of eating meals. All the subjects made no error in operation during the drinking water task. It is suggested from the experimental results that the estimated plastic bottle's height coordinate using the proposed image processing method would be acceptable for grasping it. We have developed the controllers for assistive devices, e.g., head-controlled input device [13], eye-controlled input device [14], and single finger controlled user interface [15]. Connecting the assistive robotic arm with each device, preliminary experiments were done. The results showed that these controllers are applicable to control the assistive robotic arm.

It is possible effectively and efficiently to perform image processing with a typical microcontroller-based board such as a BeagleBone Black, or a Raspberry Pi in recent years. Substituting a microcontroller-based board for the notebook computer, which is our ongoing study, will lead to less power and more realistic assistive robotic arm system. A feeding assistance with assistive robotics is the most desired assistance for people with disabilities [1]. We could not complete a meal using our previous robotic arm system [10], because any foods on the table were not searched during the meal. An optimum control for eating task with the assistive robotic arm system should be needed for our future work. In order to deal with this, we need further considerations of both position coordinates calculations of any foods and how to reach to them to pick up. The proposed vision-based method in this paper would be applicable to the former coordinate calculations.

Table 2: Averaged errors in position coordinates on each axis

	***x*-axis [mm]**	***y*-axis [mm]**	***z*-axis [mm]**
Mean	2.2	6.2	-1.3
SD	2.9	1.3	3.9

SD: standard deviation

6 CONCLUSION AND FUTURE WORK

This paper described the demonstration of the assistive robotic arm. We built the robotic arm that could firmly grasp and hold up the plastic bottle of 500 ml. It was found that the plastic bottle's position coordinates with the acceptable errors can be obtained using the vision-based assistive robotic arm system. Further experiment with people with disabilities will be needed for the future work.

ACKNOWLEDGMENT

The authors would like to thank T. Soken, M. Kuniyoshi, and anonymous referees for their valuable contributions to the progress of our work. This work was supported in part by JSPS KAKENHI, Grant-in-Aid for Scientific Research (C): Grant 22500509 and 25350671.

REFERENCES

[1]. Miller, D.P., *Asssitive robotics: An overview.* Assistive Technology and AI, 1998. p. 126-136.

[1]. Hillman, M., *Rehabilitation robotics from past to present —A historical perspective*, in *Rehabilitation Robotics*, *2003. Proceedings. 2003 International Conference on*.

[2]. Driessen, M.J.F, Kate, T.K.T., Liefhebber, F., Versluis, A.H.G., and van Woerden, J.A., *Collaborative control of the manus manipulator.* Universal Access in the Information Society, 2005. **4**(2): p. 165-173.

[3]. Mahoney, R.M., *The raptor wheelchair robot system*. Integration of Assistive Technology in the Information Age, 2001. p. 135-141.

[4]. Hillman, M., Hagan, K., Hagan, S., Jepson, J., and Orpwood, R., *The western wheelchair mounted assistive robot —the design story.* Robotica, 2002. **20**: p. 125-132.

[5]. Topping, M.J., and Smith, J.K., *The development of handy1. A robotic system to assist the severely disabled*. J. Technology and Disability, 1999. **10**(2): p. 95-105.

[6]. Ishii,S., *Meal-assistance robot "My Spoon"*. Journal of Robotics Soc. Japan. 2003. **21**(4): p. 378-381.

[7]. Ishii, S. and Arai, B., *Robot development in the field of welfare*. Journal of Robotics Soc. Japan. 2006. **24**(3): p. 304-307.

[8]. Uehara, H., Higa, H., and Soken, T., *A mobile robotic arm for people with severe disabilities*. in *IEEE BioRob*, *2010. Proceedings. 2010 International Conference on*.

[9]. Uehara, H., Higa, H., Soken, T., and Namihira, Y., *Trial development of a mobile feeding assistive robotic arm for people with physical disabilities of the extremities*, IEEJ Trans., 2011. **131**(10): p. 1-8.

[10]. Craig, J.J., *Introduction to robotics: mechanics and control.* 2nd edition, 1989, Addison-Wesley.

[11]. Bradski, G. and Kaehler, A., *Learning openCV*, 1st edition, 2008, O'Reilly.

[12]. Higa, H., Nakamura, I., and Hoshimiya, N., *A basic study on control command input device using head movement for FES system —Availability of acceleration sensors*. IEICE Trans., 2004. E87-A(6): p. 1441-1445.

[13]. Higa, H., Mihara, K., Dojo, T., Uehara, H., Kanoh, S., and Hoshimiya, N., *A video-based control command input device for FES system.* in *IEEE BioCAS, 2007. Proceedings. 2007 International Conference on*.

[14]. Higa, H., Dojo, T., Mihara, K., Uehara, H., and Asari, V.K., *A video-based user interface for people with disabilities of the fingers.* in *IEEE IACSIT-SC, 2009. Proceedings. 2009 International Conference on*.

SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
VOLUME 2, ISSUE 4
ISSN: 2054 - 7390

A novel approach to decision making of Mined Data using Dynamic Snapshot Pattern Recognition Algorithm (DS-PRA)

Mahmoud Z. Iskandarani
Faculty of Science and Information Technology, Al-Zaytoonah University of Jordan, Amman-Jordan;
m.iskandarani@hotmail.com

ABSTRACT

A new approach to pattern recognition and decision machines profiling is proposed, proved and tested. The technique adopts the Snapshot method dynamically as a function of both organization policy and the organization department's policies. Such policies are associated with individual products and services provided by the organization with departments policies derived from general organization profile with the organization policy being a function of the various department's profiles. It is proved through real data the ability of such algorithm to classify, detect and predict policy changes and identify differences between different organizations. Also, such algorithm combines the concepts of general Artificial intelligence through the use of knowledge bases and Neural Networks by utilizing a similar weights matrix.

Keywords: Snapshot, Pattern Recognition, classification, Data Mining, Intelligent Systems.

1 INTRODUCTION

In many real-world data mining tasks, decision making may change as the time changes. This is due to change in the learned knowledge as a function of data exposure with a resulting optimization of the knowledge base (KB) as a function of experience and multi-case handling, which is a function of the everyday expansion of learned knowledge [1-3].

In order to learn and make accurate predictions, the evolution of the decision making concept must be considered, and thus, a series of data sets collected at different times is needed to keep updating the data base of learned knowledge, resulting in preserving the main characteristics of the prediction model with modifications, adjustments and optimization due to the constant input and feedback to and from the knowledge base. So a snapshot of the characteristics and features of the system under consideration would suffice to reach a decision. This would save significant time and storage and it will support the decision making process without the need to have an updated data tapped in from the knowledge base [4-6].

DOI: 10.14738/tmlai.24.298
Publication Date: 3rd Aug 2014
URL: http://dx.doi.org/10.14738/tmlai.24.298

In this work it is shown that using the DS-PRA setup, a somewhat more natural way of temporal dependent data classification can be achieved. In addition this setup, can successfully replace all those processes of traditional classification scheme, such as, feature extraction and finding sufficient classification algorithms. It is also demonstrated here that this task can be done with a relatively small amount of data, which can be regarded as more suitable for real-time applications Moreover the model has the ability to convert a human dependent task of the selection of features into an automated computational task of finding DS-PRA parameters [7-10].

2 MATERIALS AND METHODS

The traditional methods of actual classification schemes [11-13], usually consist of the following basic parts as shown in Figure 1:

1) Feature Extraction
2) Applying classification algorithm
3) Decision Algorithm.

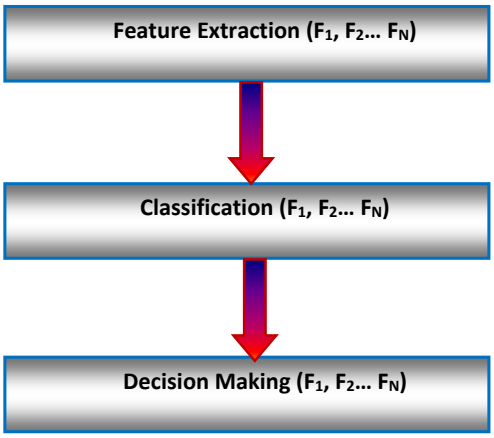


Figure 1: Traditional Decision Making System

The goal of the three steps to reduce the data explosion that is derived from real life data.

The feature extraction part requires substantial experience with the specific data task and the success of finding good features can be affected by experience, extensive analysis of the data and time-frequency manipulations. All of these are time demanding processes, and usually computationally ineffective and somewhat not natural. Their main goal is data dimensionality reduction for the classification algorithms.

The classification and decision making strictly depends on the previous one. Successful feature extraction process will result in efficient and fast learning, which affect its accuracy and generalization.

An accurate and effective model is developed from a single snapshot of data with the help of domain knowledge. The correlation of snapshots gathered over a period of time is used to establish a pattern of the organization and its decision making policy. Any change in the periodical updating of the organization decision making process will result in a change in its last snapshot and in the overall pattern and behavior of the model used.

In principle, an ensemble of data, called snapshots, collected from the database. The algorithm is then used to produce a set of basic elements that can span the original snapshot collection. It is such capability that allows extracting the representative characteristics of a decision making system of an organization. The resulting few elements that form the final profile can be regarded as dominant patterns. The proposed model and method indicate that the DS-PRA method can successfully detect pattern changes and predict decisions without the need to tap back into the database. The decision making machine profiling is affected by new data which acts as a stimulus with a dynamic continuously changing history with sliding threshold for decision making. This system is proposed and designed with an objective of achieving both system accuracy while maintaining good generalizability properties [14-18]. The overall system is illustrated in Figure 2.

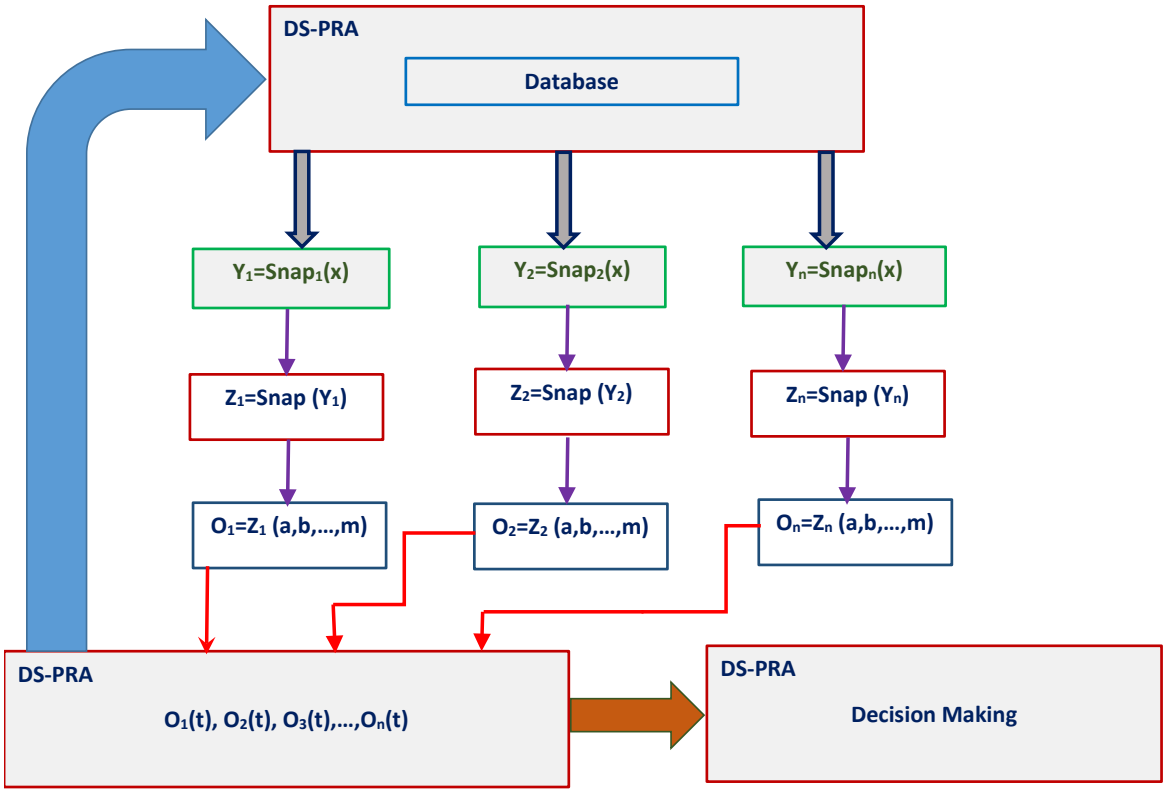


Figure 2: DS-PRA System

The resulting decisions and limits should correlate to the original policy of the organization and can be described by equation 1:

$$Decision(t_n) = f\left(Snapshot_{\ Company\ Policy}(t_1, t_2, ..., t_n)\right) ...(1)$$

With;

$$Snapshot_{\ Company\ Policy} = f\left(company\ Policy\ Variables\right) ...(2)$$

For a snapshot to be valid and utilized for decision making and prediction, equation (3) must be satisfied:

$$\frac{d(Snapshot)}{dt} = 0 ...(3)$$

The expression in (3) is used to also determine if there is a change in the decision policy if the result does not equate to zero. This indicate a change in transformation function of the concerned variables and is used to form new Snapshot if sufficient period of time is allowed to establish stability. From the previous, equation (4) is used to map the Snapshot change and to linearly predict the outcome for the new policy.

$$Var_{Policy}\left|\frac{d(Snapshot_{n+1})}{dt} - \frac{d|Snapshot_n|}{dt}\right| \leq \alpha \quad ...(4)$$

$$0 \leq \alpha \leq 1$$

For the prediction and decision making to be valid, the Snapshot variation for real and predicted data should not exceed the Snapshot variation limit of the general company policy and profile, hence:

$$Var_{data} \leq Var_{Policy} \leq \alpha ...(5)$$

3 RESULTS

Tables 1 and 2 show DS-PRA Snapshot results of policies over two years for two organizations dealing with products and services, where the transactions on these products and services are affected by the organization policy that covers compensation aspects through different departments.

Each main variable have snapshots of its sub variable that contributed to its value. The snapshot of the policy is based on historical data gathering and intelligent correlation between the snapshot variables shown in tables 1 and 2.

Table 1: Original snapshots for Organization1

$Organization_1$		
Snapshot Variable	**Policy-2012**	**Policy-2013**
Maximum Compensation	52.43%	53.13%
Minimum Compensation	40.48%	37.48%
No Compensation	6.14%	6.65%
Rejection	0.95%	2.74%
$Department_1$		
Maximum Compensation	0.75%	0.14%
Minimum Compensation	88.02%	88.54%
No Compensation	4.15%	6.57%
Rejection	7.08%	4.75%
$Department_2$		
Maximum Compensation	86.67%	81.53%
Minimum Compensation	8.04%	12.97%
No Compensation	5.23%	4.09%
Rejection	0.06%	1.40%

Table 2: Original snapshots for Organization2

$Organization_1$		
Snapshot Variable	**Policy-2012**	**Policy-2013**
Maximum Compensation	46.84%	44.63%
Minimum Compensation	47.24%	43.12%
No Compensation	5.63%	10.50%
Rejection	0.29%	1.75%
$Department_1$		
Maximum Compensation	0.67%	0.84%
Minimum Compensation	89.60%	88.83%
No Compensation	8.72%	7.26%
Rejection	1.01%	3.07%
$Department_2$		
Maximum Compensation	89.73%	86.41%
Minimum Compensation	4.83%	4.80%
No Compensation	5.29%	6.95%
Rejection	0.15%	1.84%

4 ANALYSIS AND DISCUSSION

The variables are integrated into to general variables as demonstrated in tables 3 and 4. After integration, a boundary is constructed around the integrated variables to establish center points and limits. The limits are then applied to departments in each organization to enable decision making and prediction.

Table 3: Derived Snapshots for Organization1

Organiztion1 Decision System	Policy Limits-Year$_1$	Policy Limits-Year$_2$	Var
Compensation	90.58%-92.91%	90.61%-92.93%	0.01%
No Compensation	9.42%-7.09%	9.39%-7.07%	
Department$_1$			
Compensation	88.68%-88.77%	88.68%-88.77%	0%
No Compensation	11.32%-11.23%	11.32%-11.23%	
Department$_2$			
Compensation	94.59%-94.71%	94.51%-94.62%	0.01%
No Compensation	5.41%-5.29%	5.49%-5.38%	

Table 4: Derived Snapshots for Organization2

Organiztion2 Decision system	Policy Limits-Year$_1$	Policy Limits-Year$_2$	Var
Compensation	89.39%-94.08%	87.75%-93.17%	0.73%
No Compensation	10.61%-5.92%	12.25%-6.83%	
Department$_1$			
Compensation	87.59%-90.27%	89.67%-91.9%	0.45%
No Compensation	12.41%-9.73%	10.33%-8.1%	
Department$_2$			
Compensation	91.69%-94.56%	91.21%-94.25%	0.17%
No Compensation	8.31%-5.44%	8.79%-5.75%	

From Tables 3 and 4 and referring to equations 3, 4, and 5, the following is realized:

1. The Var for each considered department in the considered organizations is within the proposed limits and verifies the validity of the algorithm.
2. Organization$_1$ shows no or negligible snapshot change over two years, with one department policy suffers no change.
3. Organization$_2$ shows more sizable shift in the general Snapshot for the organization policy with all departments suffering Sub-Snapshot change over two years.

4. All department within both organizations have their Var_{data} that represents the departments performance in providing services Less or equal to Var_{Policy}. This satisfies the proposed algorithm and prove its validity.

5. Matrix of Sub-Snapshots is formed. This is equivalent to the matrix of weights in the Neural Network algorithms, but much simpler as shown in matrices 1 and 2 shown in expressions (6) and (7).

$$M_{Organization\,1} = \begin{bmatrix} 2.33 & 2.32 \\ 0.09 & 0.09 \\ 0.12 & 0.11 \end{bmatrix} ...(6)$$

$$M_{Organization\,2} = \begin{bmatrix} 4.69 & 5.42 \\ 2.68 & 2.23 \\ 2.87 & 3.04 \end{bmatrix} ...(7)$$

From the matrices it is realized that $Organization_1$ had little or no change in its services policy. This is supported by small variation per year compared to $Organization_2$. The weights or Sub-Snapshots are used by the algorithm to predict outcomes and make decisions dynamically as Sub-Snapshots are updated. Hence the weights or Sub-Snapshots and subsequently the matrices will be further populated and can only be limited by the choice of period of time determined by the algorithm.

The general populated and accumulated matrix for an organization is given in equation (8).

$$M_{Organization\ K} = \begin{bmatrix} OrgVar_{Year_1} & OrgVar_{Year_2} & OrgVar_{Year_3} & ..... & OrgVar_{Year_n} \\ Dep1Var_{Year_1} & Dep1Var_{Year_2} & Dep1Var_{Year_3} & ..... & DepVar_{Year_n} \\ Dep2Var_{Year_1} & Dep2Var_{Year_2} & Dep2Var_{Year_3} & ..... & Dep2Var_{Year_n} \\ DepiVar_{Year_1} & DepiVar_{Year_2} & DepiVar_{Year_3} & ..... & DepiVar_{Year_n} \\ DepnVar_{Year_1} & DepnVar_{Year_2} & DepnVar_{Year_3} & ..... & DepnVar_{Year_n} \end{bmatrix} ...(8)$$

where: Org: Organization, Dep: Department.

The Var matrix is formed as shown in equation (9) and (10).

$$Var_{\ Organization\,1} = \begin{bmatrix} 0.00 \\ 0.01 \end{bmatrix} \leq 0.01 \leq 1 ...(9)$$

$$Var_{\ Organization\ 2} = \begin{bmatrix} 0.45 \\ 0.17 \end{bmatrix} \leq 0.73 \leq 1 \ldots(10)$$

The Var matrix changes over time in an accumulative way. Equation (11) describes a general form of such matrix.

$$Var_{\ Organization\ K} = \begin{bmatrix} Var_1 \\ Var_2 \\ Var_3 \\ Var_i \\ Var_n \end{bmatrix} \leq 1 \ldots(11)$$

From the Var equations, the rest of the Var values for other departments within the organization can be predicted as it is restricted to a maximum value of 1. So, for $Organization_1$, it is clear that its policy is constant with hardly any variation, which leaves 0.99 span for overall policy change and for adding other departments or variables, and 0 margin of change for current Var_{Total}. For $Organization_2$, and as departments1 and 2 consumed most of the Var values, the span for the rest of departments is narrow and of value 0.27 to reach the maximum allowed value of 1 and 0.11 for current Var_{Total}. Such approach allows to predict and plan policies that reflect on outcomes by using parts of the organization's profile and policies and project that to obtain a dynamic curve of behavior. the algorithm is further enhanced through scheduled policy selection, hence, a variable that is selected now, will become constant next iteration and by final correlation and integration, the whole picture becomes integrated. this can be illustrated in equation (12) and (13).

$$Var_{Total} = \sum_{i=1}^{n} Var_i \ldots(12)$$

$$Snapshot_{Total} = f(Var_{Total}) \ldots(13)$$

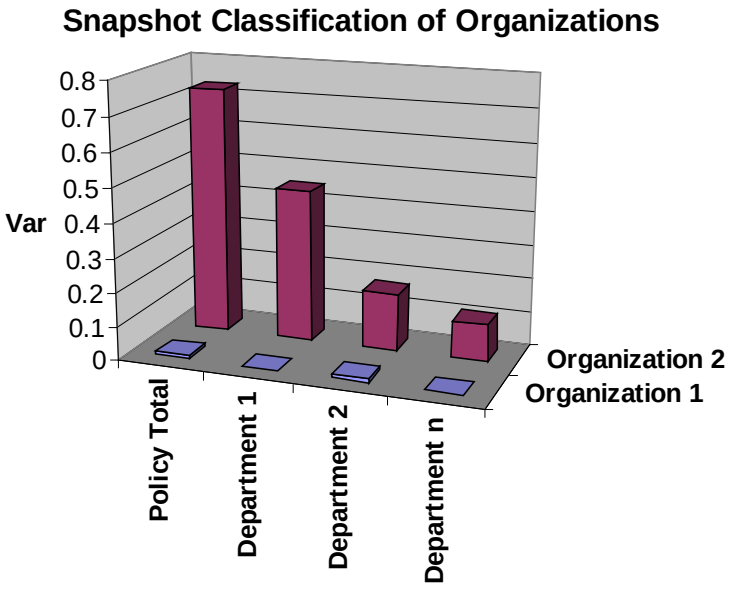


Figure 3: Classification of Two Organizations

Figure 3 shows a clear difference in two organizations policies. These organizations deals with same products but have different policy change rates. Obviously organization 1 has a constant and stable policy, which should reflect on its market dealings and its services in comparison to organization 2. such values obtained through Snapshot algorithm reflects on two main issues:

1) Confidence level in the organization if stability is important.
2) Flexibility in policy adjustment to suit market and people, if such changes are for the consumer benefits.

5 CONCLUSION

The developed algorithm proved to be effective in:

1. Reducing data size that characterizes general decision making of an organization based on historical data.
2. Enable decision making and prediction dynamically.
3. Provide and efficient method for change in decision making as a function of policy change.
4. Enable future planning of any organization policy and strategy.

REFERENCES

[1]. J. Ramon, C. comendant, Open Problem: Learning Dynamic Network Models from a Static Snapshot. 25th Annual Conference on Learning Theory, JMLR: Workshop and Conference Proceedings, 2012. 23: p. 45.1-45.3.

[2]. Z. Mousavinasab, H. Bahadori, AN OVERVIEW ON DATA MODELS FOR KNOWLEDGE DISCOVERY FROM DATABASES. International Journal of Advanced Research in IT and Engineering, London:2013. 2(7): p.12-20.

[3]. G. Kou, W.Wu, TAn Analytic Hierarchy Model for Classification Algorithms Selection in Credit Risk Analysis. Mathematical Problems in Engineering, 2014. 2014(297563): p. 1-7.

[4]. S. Beniwal, J. Arora, Classification and Feature Selection Techniques in Data Mining. International Journal of Engineering Research & Technology (IJERT), 2012. 1(6): p. 1-6.

[5]. A. Moeinian, S. Baladehi, A. Zolfagharian, Hybrid Genetic Algorithm Using the Solving Open Shop Scheduling. International Journal of Engineering Research and Technology, 2013. 5(2): p. 1-10.

[6]. V. Vasani, R. Gawali, Classification and performance evaluation using data mining algorithms. International Journal of Innovative Research in Science, Engineering and Technology, 2014. 3(3): p. 10453-10458.

[7]. R. Kumar, R. Verma, Classification Algorithms for Data Mining:A Survey. International Journal of Innovations in Engineering and Technology (IJIET), 2012, 1(2): p. 7-14.

[8]. R. Lokeshkumar, P. Sengottuvelan, M. Vina, An Approach for Web Personalization using Relational Based Fuzzy Clustering Ontology Model, Biomedical Engineering, Australian Journal of Basic and Applied Sciences, 2014. 8(2): p. 18-22.

[9]. M. Peker, B. Sen, S. Bayir. Using Artificial Intelligence Techniques for Large Scale SetPartitioning Problems. Procedia Technology, 2012 , 1: p. 44 – 49.

[10]. N. Davuth, K. Sung-Ryul, Classification of Malicious Domain Names using Support Vector Machine and Bi-gram Method, International Journal of Security and Its Applications, 2013, 7(1): p. 51-58.

[11]. H. Yang, S. Fong, G. Sun, R. Wong, A Very Fast Decision Tree Algorithm for Real-Time Data Mining of Imperfect Data Streams in a DistributedWireless Sensor Network. International Journal of Distributed Sensor Networks, 2012, 2012(863545): p.1-16.

[12]. S. Dandu, B. Deekshatulu, P. Chandra, Improved Algorithm for Frequent Item sets Mining Based on Apriori and FP-TreeIEEE. Global Journal of Computer Science and Technology Software & Data Engineering, 2013. 13(2): p. 1-5.

[13]. Y. Yang, Z. Zhi-Hua, A Framework for Modeling Positive Class Expansion with Single Snapshot. Springer-Verlag Berlin Heidelberg, 2008, p. 429-440.

[14]. L. Zhenliang, B. Wang, X. Xiaowei, P. Hannam, Environmental emergency decision support system based on Artificial Neural Network. Safety Science, 2012. 50(2012): p. 150-163.

[15]. S. Gupta, D. Kumar, A. Sharma, DATA MINING CLASSIFICATION TECHNIQUES APPLIED FOR BREAST CANCER DIAGNOSIS AND PROGNOSIS. Indian Journal of Computer Science and Engineering (IJCSE), 2011. 2(2): p. 188-195.

[16]. X. Wu, et al., Top 10 algorithms in data mining. Knowl Inf Syst, 2008, 14: p.1–37.

[17]. S. Strohmeier, F. Piazza, Domain driven data mining in human resource management: A review of current research, Expert Systems with Applications, 2013, 40(7):p.2410-2420.

[18]. H. Tsai, Knowledge management vs. data mining: Research trend, forecast and citation approach. Expert Systems with Applications, 2013, 40(8): p. 3160-3173.

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 2, ISSUE 4
ISSN: 2054 - 7390

A Machine Learning Approach for Prediction of Gibberellic Acid Metabolic Enzymes in Monocotyledonous Plants

Sreepriya P.[1], Naganeeswaran S. [1], Hemalatha N. [2], Sreejisha P. [1] and Rajesh M. K.[1,*]
[1]*Bioinformatics Centre, Central Plantation Crops Research Institute, Kasaragod, Kerala, India*
[2]*AIMIT, St. Aloysius College, Mangalore, Karnataka, India*
sreepriya.pradeep@gmail.com, naganeeswaran@gmail.com, hemasree71@gmail.com, sreeji279@gmail.com, mkraju.cpcri@gmail.com
[*Corresponding author: Phone: +91-4994-232894-284; Fax: +91-4994-232322]

ABSTRACT

Gibberellins (GA) are one of the most important phytohormones that control different aspects of plant growth and influence various developments such as seed germination, stem elongation and floral induction. More than 130 GAs have been identified; however, only a small number of them are biologically active. In this study, five enzymes in GA metabolic pathway in monocots have been thoroughly researched namely, ent-copalyl-diphosphate synthase (CPS), ent-kaurene synthase (KS), ent-kaurene oxidase (KO), GA 20-oxidase (GA20ox), and GA 2-oxidase (GA2ox). We have designed and implemented a high performance prediction tool for these enzymes using machine learning algorithms. 'GAPred' is a web-based system to provide a comprehensive collection of enzymes in GA metabolic pathway and a systematic framework for the analysis of these enzymes for monocots. WEKA-based classifiers (Naïve-Bayes) and Support Vector Machine (SVM) based-modules were developed using dipeptide composition and high accuracies were obtained. In addition, BLAST and Hidden Markov Model (HMMER-based model) were also developed for searching sequence databases for homolog's of enzymes of GA metabolic pathway, and for making protein sequence alignments.

Keywords: GA, SVM, WEKA, BLAST, HMMER

1 INTRODUCTION

Gibberellic acids (GA) are naturally occurring phytohormones that regulate growth and influence various developmental processes, including stem elongation, germination, dormancy, flowering, enzyme induction, and leaf and fruit senescence [1]. They are also involved in the discernment of environmental stimuli, thus are significant not only for a plant's growth and development but also in awareness of its environment. Gibberellins are diterpenoid acids which

DOI: 10.14738/tmlai.24.375
Publication Date: 28th August 2014
URL: http://dx.doi.org/10.14738/tmlai.24.375

are formed by the terpenoid pathway in plastids and then modifying the endoplasmic reticulum (ER) and cytosol until they are biologically-active [2]. Gibberellins are derived by the *ent*-gibberellane skeleton, but are synthesized by *ent*-kaurene [3]. The GA biosynthetic pathway can be divided into three stages, each stage residing in a different cellular compartment *viz*. plastid, the endoplasmic reticulum, and the cytosol [4].

A number of experimental studies have explained thoroughly the biosynthetic functions of gibberellic acid. In this study, five enzymes involved in GA metabolic pathway in monocots have been thoroughly researched namely, *ent*-copalyl-diphosphate synthase (CPS), *ent*-kaurene synthase (KS), *ent*-kaurene oxidase (KO), GA 20-oxidase (GA20ox), and GA 2-oxidase (GA2ox) [5]. In this study, we have designed and implemented a high performance prediction tool based on kernel-based Machine Learning Algorithms *viz*., Support Vector Machine (SVM) and WEKA for prediction of enzymes in gibberellic acid metabolic pathway. In addition, standalone BLAST and Hidden Markov Model (HMMER-based model) were also developed for searching sequence databases for homolog's of enzymes of GA metabolic pathway, and for making protein sequence alignments. 'GAPred' was developed using the evolutionary and sequence features of a protein sequence and the performance of the each model was evaluated using cross-validation techniques. Based on our study, we have also created and hosted a web server for predicting enzymes involved in GA metabolism.

2 MATERIALS AND METHODS

2.1 Dataset

In the present study, two datasets were considered for the development of the prediction tool 'GAPred'. Positive (+ve) dataset comprised of 102 selected GA metabolic enzyme protein sequences from monocots *viz*., date palm (*Phoenix dactylifera*), coconut (*Cocos nucifera*), rice (*Oryza sativa*), barley (*Hordeum vulgare*), maize (*Zea mays*), banana (*Musa acuminata*), and brachypodium (*Brachypodium distachyon*), after redundancy elimination by using ClustalW. Similarly negative (-ve) dataset was created by using same numbers of non-GA metabolic enzymes sequences. The sequences were retrieved from NCBI in FASTA format (http://www.ncbi.nlm.nih.gov/). Domains of enzymes involved in GA metabolic pathway were identified using Pfam search and PRINTS search and most of the identified enzyme domains are known to be conserved in related species. To avoid the over estimation, we clustered the protein sequences from positive data (+ve) set with a threshold of 30% identity by CD-HIT (Cluster Database at High Identity with Tolerance). Out of 102 GA metabolic enzymes sequence, 62 proteins were randomly selected for the creation of training set. Similarly training set of non-GA metabolic enzymes sequence was created. To test the reliability of the prediction tool, we also prepared a test set of 40 GA metabolic enzymes sequences and non-GA metabolic enzymes sequences which were not the part of training set.

2.2 Support Vector Machine

Support Vector Machines (SVM) are a group of rapid optimization machine learning algorithms with strong theoretical foundation, which have been used for many kinds of pattern recognition [6-8]. SVMs are now extensively used for biological applications and methods such as classifying objects as diverse as protein and DNA sequences, mass spectra and microarray expression profiles [9]. In this work, SVM has been implemented by using $SVM^{multiclass}$ package [10] which possesses two modules: SVM_multiclass_learn and SVM_multiclass_classify. The first module (SVM_multiclass_learn) is concerned for preparing models learned from the training dataset (+ve and -ve) and the final one classifies the data by using the models prepared by SVM_multiclass_learn. Here, we have trained the $SVM^{multiclass}$ by using a set of positive and negative datasets, and produces a model (classifier) that can be used to identify the potential enzymes involved in gibberellic acid metabolic pathway. With the help of this package the user can select various kernel functions (linear, polynomial, radial basis, sigmoid or any other user defined kernel) for preparing models. In SVMs, the kernel function selected must be the most favorable one. Here in the creation of SVM models, we have used three types of kernel functions: linear, polynomial, and radial. The performance of SVM based methods has been optimized by regulating SVM parameters so that maximum accuracy could be obtained.

2.3 WEKA

WEKA stands for 'Waikato Environment for Knowledge Analysis' and is a free open source software developed by at the University of Waikato, New Zealand. This popular machine learning software contains a collection of algorithms and visualization tools for data analysis, analytical modeling and also graphical user interfaces for easy access to this functionality. In the given work, we used WEKA classification [11], where different attributes of a protein sequence are analyzed to classify the protein sequence into one of the predefined classes. Both train and test set was used to get the classification of the data set by using better algorithms. The performance of WEKA has been optimized by tuning evaluation parameters and visualization schemes, in order to analyze the accuracy of classifiers.

2.4 Sequence Features

Dipeptide composition gives comprehensive information about each protein sequence that possess sequence feature. Generally, the total number of amino acids is 20 and thus the theoretical number of possible dipeptides is 400. A matrix of these 400 dipeptides was generated for each protein and is then given as an input to both SVM and WEKA. Each dipeptide frequency is calculated by the formula: $DF_{ij}=N_{ij}/N$ where, N_{ij}=count of the ijth dipeptide; N=total number of possible dipeptides; i, j = 1-20 amino acid.

2.5 Performance Assessment of GAPred

With the help of statistical calculations, we generally examine the efficiency of a predictor either using single independent dataset test, cross-validation test or jackknife test. However, jackknife test method takes much longer time to examine a predictor based on SVM and WEKA [12] and therefore, in this work, we have adopted 10-fold cross-validation for WEKA and 5-fold cross-validation for SVM^{multiclass} and independent data set validation techniques were adopted for measuring performance. In 10-fold cross-validation test, the significant dataset was divided randomly into ten equally sized sets. The training and testing methods were carried out ten times with each individual set used for testing and for the nine sets left behind for training. Similarly in 5-fold cross validation, the dataset was partitioned randomly into five equally sized sets. In the independent dataset test, the training dataset used to train the predictor does not contain any data that is to be tested.

2.6 Evaluation Parameters

We had made use of five parameters to evaluate the reliability of the prediction tool, they are: Accuracy (Ac), Sensitivity (Sn), Specificity (Sp), Precision (Pr) and Matthew's Correlation Coefficient (MCC). Accuracy defines the proportion of correctly predicted proteins (Eq.1). The sensitivity (Sn) and specificity (Sp) represent the correct prediction ratios of positive (+ve) and negative data (-ve) sets of metabolic enzymes of gibberellic acid sequence respectively (Eq. 2 and 3). Precision is the proportion of the predicted positive cases that were correct (Eq.4). Matthew's correlation coefficient or MCC [13-14] is a statistical parameter which also used to estimate the accuracy of prediction (Eq.5). MCC may range from -1 to +1 and the highest MCC value indicates better prediction [15].

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (1)$$

$$Sensitivity = \frac{TP}{TP+FN} \times 100 \quad (2)$$

$$Specificity = \frac{TN}{FP+TN} \times 100 \quad (3)$$

$$Precision = \frac{TP}{TP+FP} \times 100 \quad (4)$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (5)$$

Where TP=number of true positives; TN=number of true negatives; FP=number of false positives; FN=number of false negatives. In this work, metabolic enzymes of GA sequences are true positives and non- metabolic enzymes of GA sequences are true negatives.

2.6 Sequence Similarity search using standalone BLAST

The standalone BLAST programs are freely provided as open-source software by NCBI. With stand-alone BLAST we can made our own databases to search against. In this study, sequences were searched against the protein non-redundant (nr) database in association with standalone BLASTp and to detect homology of metabolic enzymes of gibberellic acid proteins and result was analysed.

2.7 Sequence Similarity Search using HMMER

Profile Hidden Markov models (profile HMMs) techniques are one of the most dominant methods for protein homology detection [16]. HMMER helps to find out protein sequences which are similar in sequence databases and to make protein sequence alignments. HMMER becomes particularly powerful when the query is a multiple sequence alignment of a sequence family rather than for single query sequences. It makes a profile of the query that assigns a position-specific scoring system for substitutions, insertions, and deletions [17]. HMMER profiles are probabilistic models called “profile Hidden Markov models” (profile HMMs). Because of the strength of its underlying probability models, HMMER aims to be much more accurate and more capable of finding out remote homolog’s rather than BLAST, FASTA or any of the other sequence alignment and database search tools based on older scoring methodology [18]. Hence, in this study, we have used HMMER to detect homology of metabolic enzymes of gibberellic acid proteins and a remarkable result was analyzed.

2.8 ROC Curves

By making use of ROC curves, a graph created by plotting the fraction of false positives (FPR) against true positives (TPR) at various threshold settings [19], we can explain the performance of multi class classifiers in SVM and WEKA more specifically. TPR is also known as sensitivity, and FPR is 1-specificity or true negative rate. ROC analysis is linked in a direct and natural method to benefit analysis of diagnostic decision making. ROC curves useful for the evaluation of machine learning techniques and data mining research.

2.9 Web-server

We have implemented the prediction tool “GAPred” in a web server. The program is written entirely in HTML, PHP and PERL program in a Linux platform. The tool page serve as the platform for submitting data where users can either paste or upload sequence which should be in standard FASTA format. It also provides a comprehensive collection of enzymes in GA metabolic pathway and introduces a user to gibberellic acid metabolic pathway.

3 GAPRED

3.1 Evaluation of Performance of GAPred

We have carried out 10-fold cross-validation for WEKA and 5-fold cross-validation for SVM and also independent data test validation to evaluate the performance of GAPred (Tables 1-4). Cross validation and independent data test results for SVM from the Tables 1 and 3 shows that cross validation has better result for dipeptide composition methods compared to independent data test. While in the case of WEKA, the independent data set has better result than cross validation method.

3.2 Comparison of GAPred with BLAST and HMMER

We have also used standalone BLAST and HMMER to detect homology of metabolic enzymes of gibberellic acid. This was used to compare an input protein sequence with a created database to generate the homology of the given sequence. A comparison of enzyme proteins was conducted with standalone BLAST and HMMER database and an accuracy of 99% and 93% were obtained (Tables 5-6). By making a comparison between SVM, WEKA, BLAST and HMMER from Table 7 and Figure 1, it can be seen that SVM has achieved 100% accuracy and MCC value equal to 1 that an ideal classification method should possess. Hence, SVM was selected to be the best model for GAPred.

Table 1 Validation of independent data test results of dipeptide composition of metabolic enzymes of gibberellic acid with SVM^{multiclass}

Algorithm	Sn (%)	Sp (%)	Ac (%)	Pr (%)	MCC
Linear	5	100	53	100	0.16
Polynomial	95	100	98	100	0.95
RBF	93	95	94	95	0.88

Table 2 Validation of independent data test results of dipeptide composition of metabolic enzymes of gibberellic acid with WEKA

Classifiers	Sn (%)	Sp (%)	Ac (%)	Pr (%)	MCC
Naïve Bayes	98	100	99	100	0.98
Bayes Net	95	100	98	100	0.95
Decorate	83	100	91	100	0.84

Table 3 Comparison of the prediction performance of three kernels of SVMmulticlass with dipeptide composition technique using 5-fold cross validation

Algorithm	Sn (%)	Sp (%)	Ac (%)	Pr (%)	MCC
Linear	100	100	100	100	1
Polynomial	100	100	100	100	1
RBF	100	100	100	100	1

Table 4 Comparison of the prediction performance of three classifiers of WEKA with dipeptide composition technique using 10-fold cross validation

Classifiers	Sn (%)	Sp (%)	Ac (%)	Pr (%)	MCC
Naïve Bayes	89	100	94	100	0.89
Bayes Net	95	97	96	97	0.92
Decorate	95	95	95	95	0.90

Table 5 Comparison of the prediction performance of standalone BLAST with created database of domains of metabolic enzymes of gibberellic acid

	Sn (%)	Sp (%)	Ac (%)	Pr (%)	MCC
BLAST	100	98	99	98	0.98

Table 6 Comparison of the prediction performance of HMMER with created database of domains of metabolic enzymes of gibberellic acid

	Sn (%)	Sp (%)	Ac (%)	Pr (%)	MCC
HMMER	90	97	93	96	0.87

Table 7 Comparison of the prediction performance of three methods with metabolic enzymes of gibberellic acid sequences

	Sn (%)	Sp (%)	Ac (%)	Pr (%)	MCC
SVM	100	100	100	100	1
WEKA	98	100	99	100	0.98
BLAST	100	98	99	98	0.98
HMMER	90	97	93	96	0.87

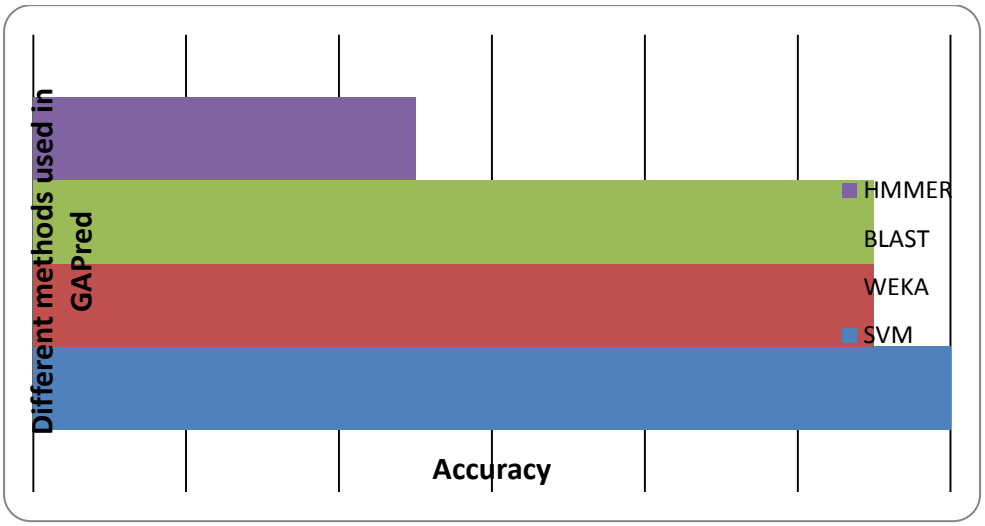


Figure 1: Comparison of performance validation of GAPred with different methods

3.3 ROC curve

We have plotted the ROC curves for SVM and WEKA based on the independent test performance of the dipeptide compositions. From the ROC curves (Figures 2-3), representing the relationship between sensitivity and (1-specificity) for a class, it is clear that the SVM composition module represents a perfect classifier since the curve obtained is an inverted 'L', which is a desirable characteristic of an ROC curve. Each point on the ROC curve was plotted based on different threshold scores.

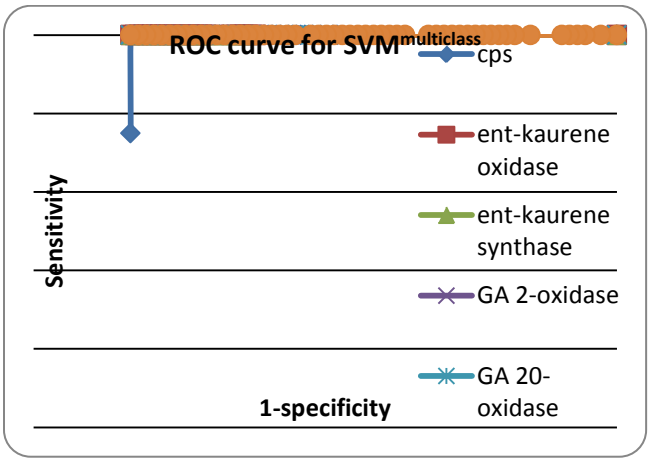


Figure 2: ROC curve for dipeptide composition in SVM^{multiclass} using independent test results

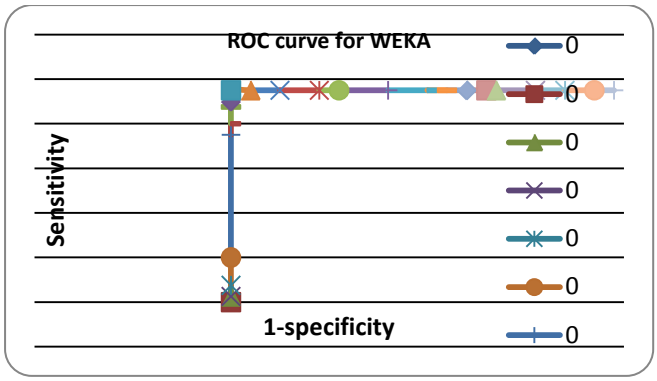


Figure 3: ROC curve for dipeptide composition in WEKA using independent test results

3.4 Description of Web Server

We have implemented the prediction tool "GAPred" in a web server. The tool was developed in PERL program and web interface in PHP and HTML to assess the user queries, in Linux platform. The tool page serve as the platform for submitting data where users can either paste or upload sequence which should be in standard FASTA format (Figure 4). It also provides a comprehensive collection of enzymes in GA metabolic pathway and introduces the user about gibberellic acid. The tool is freely available at http://gapred.cpcribioinformatics.in/gapred/

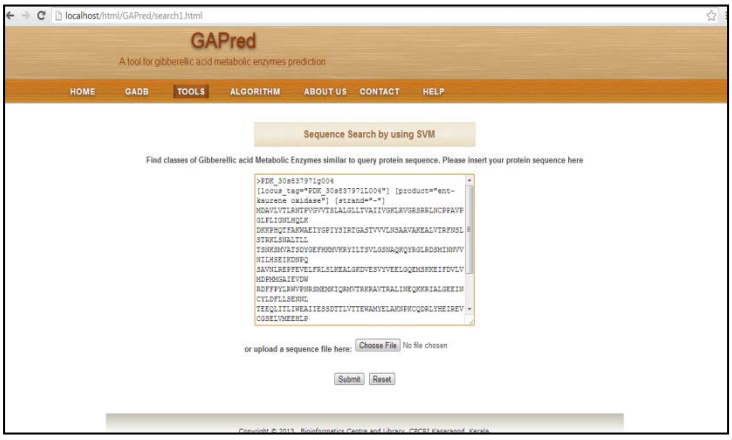


Figure 4: Web interface of GAPred

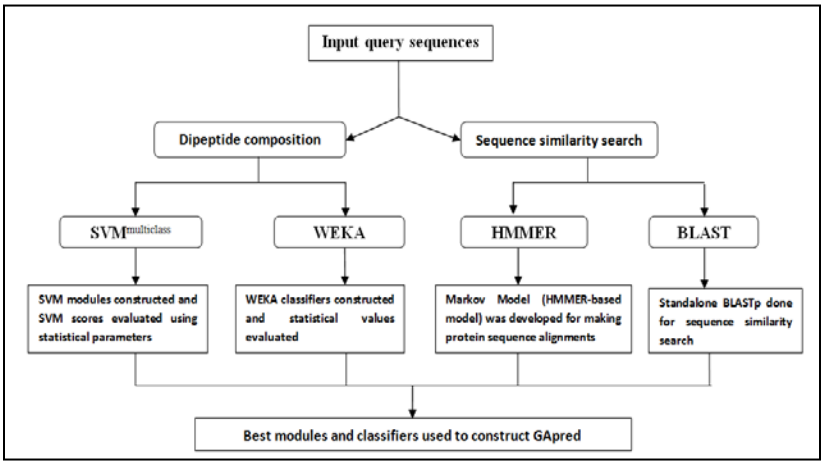


Figure 5: The architecture of the GAPred server.

4 CONCLUSION

In this work, we have described SVM and WEKA-based approaches for the prediction of enzymes in gibberellic acid metabolic pathway based on dipeptide composition. Comparison of standalone BLAST and HMMER- based homology searches with machine learning algorithms revealed that the latter performed better compared to homology-based tools. Based on kernel methods, we have developed and implemented an efficient and easy to use user-friendly prediction server called 'GAPred' for predicting five gibberellic acid metabolic enzymes. The sensitivity and specificity reaches 100% for prediction of gibberellic acid metabolic enzymes. We expect that the tool may be a useful resource for researchers as it is freely available.

REFERENCES

[1]. Phinney, B.O., The history of gibberellins. *In*: The Biochemistry and Physiology of Gibberellins, Crozier, A. (Ed.), Praeger Publishers, New York , USA , 1983. 1: p. 19–52.

[2]. Lichtenthaler, H.K., Rohmer, M. and Schwender, J., Two independent biochemical pathways for isopentenyl diphosphate biosynthesis in higher plants. Physiologia Plantarum, 1997. 101: p. 643–652.

[3]. Graebe, J.E., Gibberellin biosynthesis and control. Annual Review of Plant Physiology, 1987. 38: p. 419–465.

[4]. Chappell, J., Biochemistry and molecular biology of the isoprenoid biosynthetic pathway in plants. Annual Review of Plant Physiology and Plant Molecular Biology, 1995. 46: p. 521-547.

[5]. MacMillan, J., Biosynthesis of the gibberellin plant hormones. Natural Product Reports, 1997. 14: 221–244 .

[6]. Vapnik, V.N., An overview of statistical learning theory, Neural Networks. IEEE Transactions, 1999. 10: p. 988-999.

[7]. Cortes, C. and Vapnik, V. Support Vector Networks. Machine Learning, 1995. 20: p. 273-297.

[8]. Burges, C.J.C., A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 1998. 2: p. 121-167.

[9]. Noble, W.S., Support vector machine applications in computational biology. *In*: Kernel Methods in Computational Biology. Schoelkopf, B., Tsuda, K. and Vert, J. P. (Eds.), Cambridge, MA: MIT Press, 2004. p. 71–92.

[10]. Joachims, T., Making large-scale SVM learning practical. *In*: Advances in Kernel Methods: Support Vector Learning. Schoelkopf, B.,Burges, C. and Smola,A. (Eds.), Cambridge MA: MIT Press, 1999. p. 41–56.

[11]. Üney, F. and Türkay, M., A mixed-integer programming approach to multi-class data classification problem. European Journal of Operational Research, 2006. 173(3): p. 910-920.

[12]. Chou, K.C. and Zhang, C.T., Prediction of protein structural classes. Critical Reviews in Biochemistry and Molecular Biology, 1995. 30: p. 275–349.

[13]. Matthews, B.W., Comparison of the predicted and observed secondary structure of T4 phage lysozyme. Biochimica et Biophysica Acta, 1975. 405: p. 442-451.

[14]. Baldi ,P., Brunak, S., Chauvin, Y., Andersen, C.A.F. and Nielsen, H., Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics, 2000. 16: p. 412-424.

[15]. Carugo, O., Detailed estimation of bioinformatics prediction reliability through the Fragmented Prediction Performance Plots. BMC Bioinformatics, 2007. 8: p. 380.

[16]. Park, J., Karplus, K., Barrett, C., Hughey, R., Haussler, D., Hubbard, T. and Chothia, C., Sequence comparisons using multiple sequences detect three times as many remote homologues as pair-wise methods. Journal of Molecular Biology, 1998. 284: p. 1201-1210.

[17]. Hughey, R. and Krogh, A., Hidden Markov models for sequence analysis: extension and analysis of the basic method. Computer applications in the Biosciences, 1996. 12: p. 95-107.

[18]. Mitchison, G.J. and Durbin, R., Tree-based maximal likelihood substitution matrices and hidden Markov models. Journal of Molecular Evolution, 1995. 41: p. 1139-1151.

[19]. Swets, J.A., Measuring the accuracy of diagnostic systems. Science, 1998. 240: p. 1285–1293.

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 2, ISSUE 4
ISSN: 2054 - 7390

Classifying Documents with Poisson Mixtures

Hiroshi Ogura, Hiromi Amano, Masato Kondo
Department of Information Science, Faculty of Arts and Sciences at Fujiyoshida, Showa University, 4562 Kamiyoshida, Fujiyoshidacity, Yamanashi 403-0005, Japan;
ogura@cas.showa-u.ac.jp, kayanm@cas.showa-u.ac.jp, mkondo@nr.showa-u.ac.jp

ABSTRACT

Although the Poisson distribution and two well-known Poisson mixtures (the negative binomial and K-mixture distributions) have been utilized as tools for modeling texts for over last 15 years, the application of these distributions to build generative probabilistic text classifiers has been rarely reported and therefore the available information on applying such models to classification remains fragmentary and even contradictory. In this study, we construct generative probabilistic text classifiers with these three distributions and perform classification experiments on three standard datasets in a uniform manner to examine the performance of the classifiers. The results show that the performance is much better than that of the standard multinomial naive Bayes classifier if the normalization of document length is appropriately taken into account. Furthermore, the results show that, in contrast to our intuitive expectation, the classifier with the Poisson distribution performs best among all the examined classifiers, even though the Poisson model gives a cruder description of term occurrences in real texts than the K-mixture and negative binomial models do. A possible interpretation of the superiority of the Poisson model is given in terms of a trade-off between fit and model complexity.

Keywords: Poisson distribution, Negative binomial distribution, K-mixture distribution, Text classification, Akaike's information criterion, Bayesian information criterion

1 INTRODUCTION

The Poisson distribution is one of the most fundamental discrete distributions for describing the probability of count data (the probability of a given number of events) occurring in a fixed interval of time or space. For text modeling, the Poisson distribution is appropriate for describing the number of occurrences of a certain word in documents of fixed length when the assumption that each word occurs independently holds in an approximate sense. It has been well established, however, that the Poisson model does not fit observation data [1]. The reason for the failure of the Poisson model is that, for most words, the predicted variance, which is

DOI: 10.14738/tmlai.24.388
Publication Date: 28th August 2014
URL: http://dx.doi.org/10.14738/tmlai.24.388

equal to the Poisson mean (the expected number of occurrences during the given interval), systematically underestimates the actual variance. Although this imperfect description of word distributions by the Poisson model can be used for keyword selection in information retrieval [2] and for feature selection in text categorization [3-5], improvement of the Poisson model will inevitably be needed in various fields where word distributions are analyzed quantitatively.

As proposed by Church and Gale [1], the description by the usual Poisson distribution can be improved by extension to Poisson mixtures. Here, a Poisson mixture is a probability mass function that is expressed as a sum of finite or infinite Poisson distributions using a certain weighting function. Indeed, the K-mixture [6] and the negative binomial distributions [1], both of which are Poisson mixtures in the sense that they are expressed in the form of infinite superposition of the Poisson distribution, have been found to give a better description of the observed variance in actual documents than that of the usual Poisson, and these Poisson mixtures have been successfully utilized during the last 15 years [7-13].

In spite of the clear success of the K-mixture and negative binomial models for describing word distributions in real texts, attempts to utilize these models to construct generative probabilistic classifiers, however, have rarely been reported. To the best of our knowledge, the main studies on text classifiers using the usual Poisson model and the K-mixture and negative binomial models can be summarized as follows.

- Kim et al. [14, 15] used the Poisson distribution to build a text classifier and showed that their classifier performs much better than the multinomial naive Bayes classifier. However, since their proposed method is a sophisticated one in which additional parameter tuning is required, their classifier is not fully suitable for easy use.

- Eyheramendy et al. [16] compared the performance of four probabilistic models in text classification: the Poisson, Bernoulli, multinomial, and negative binomial models. They found that the multinomial model performs best in terms of the micro-F1 measure, and also that the Poisson and Bernoulli models are very similar in performance and are the second-best choices; the negative binomial model was found to be the worst. In short, their result showed that the usual Poisson and negative binomial models do not outperform the multinomial naive Bayes classifier.

- Airoldi et al. [17, 18] presented statistical models based on the Poisson and negative binomial distributions for text and showed that their models perform better than the widely used multinomial naive Bayes classifier in text classification tasks. The overall behavior of their classifiers indicated that the negative binomial performs best; the Poisson, the second best; and the multinomial, the worst. However, the difference in classification accuracy among the three classifiers examined was sometimes too small to judge which is the best and which is the second best, and therefore was not sufficient to

make a convincing argument that the Poisson and the negative binomial are superior to the multinomial.

The point emerging from this review of the literature is that the information on the application of the Poisson and negative binomial distributions for building generative probabilistic classifiers is still fragmentary and even contradictory. Furthermore, the application of the K-mixture model to text classifiers, which is a widely used Poisson mixture along with the negative binomial distribution, has not yet been reported.

The question motivating this study is whether the multinomial distribution embedded in the most widely used naive Bayes classifiers can be replaced with the usual Poisson, the negative binomial, or the K-mixture. The purpose of this work is therefore to show that these three models are useful tools for describing word distributions in real texts and to show the extent to which the models can be appropriately used in text classification. To determine whether these three models are useful in classification tasks, the accuracy of the proposed classifiers with the three models are examined using three standard datasets. The results lead us to conclude that these classifiers perform much better than the multinomial naive Bayes classifier does, if we construct the three classifiers with appropriate consideration of document length normalization. Another important finding is that, among the three examined classifiers, the classifier with the usual Poisson model performs best, contrary to our intuitive expectation based on the Poisson model giving a cruder description of word distributions in real texts than do the negative binomial and the K-mixture models. The origin of this better performance of the Poisson can be explained in terms of a trade-off between fit and model complexity, as will be presented later.

The rest of this paper is organized as follows. In the next section, we will describe the frameworks of the three models, (i.e., the Poisson, negative binomial, and K-mixture models) for texts, and how to construct classifiers by using these frameworks. Two different methods for normalizing document length are also described in the next section. In Section 3, we summarize our experiments on automatic text classification. Section 4 presents the results of the experiments, and in Section 5, the observed characteristics of the proposed classifiers are discussed. In Section 6, we give our conclusions and suggest directions for future investigation.

2 FORMULATION OF CLASSIFIERS

2.1 Multinomial naive Bayes

First, we briefly review the multinomial naive Bayes and some notation and symbols that will be used later. The framework described here is a standard one [19, chapter 6] and thus we use it as a reference classifier in our experiments.

The multinomial naive Bayes classifier is widely used in text categorization because it can achieve good performance in various tasks and because it is simple enough to be practically

implemented even when the number of features is large. The simplicity is due primarily to the following two assumptions. First, an individual document is assumed to be represented as a vector of word counts (bag-of-words representation). Since this representation greatly simplifies further processing, all three of the generic probabilistic classifiers investigated in this work inherit this first assumption. Next, documents are assumed to be generated by repeatedly drawing words from a fixed multinomial distribution for a given class, and word emissions are thus independent.

From the first assumption, documents can be represented as vectors of count-valued random variables. The ith document in a considered class c is then expressed as

$$d_{ci} = (x_{i1}, x_{i2}, \cdots, x_{ij}, \cdots, x_{i|V|}), \tag{1}$$

where x_{ij} is the count of the jth word t_j in the ith document belonging to class c and $|V|$ is vocabulary size; in other words, we have assumed here that the vocabulary of the considered dataset is given by $V = \{t_1, t_2, \cdots, t_{|V|}\}$ where t_j is the jth word in the vocabulary. From the second assumption, the probability of the document d_{ci} given by vector (1) is

$$p(d_{ci} \mid \theta_c) = \frac{\left(\sum_{j=1}^{|V|} x_{ij}\right)!}{\prod_{j=1}^{|V|}(x_{ij}!)} \prod_{j=1}^{|V|} \theta_{cj}^{x_{ij}}, \tag{2}$$

where θ_{cj} is the probability for the emission of t_j and is subject to the constraints $\sum_{j=1}^{|V|} \theta_{cj} = 1$. Note that for text classification, the parameters θ_{cj} must be evaluated for each possible class c. We use the estimator for θ_{cj} given by

$$\hat{\theta}_{cj} = \frac{1 + \sum_{i=1}^{|D_c|} x_{ij}}{|\ V\ | + \sum_{i=1}^{|D_c|} \sum_{j=1}^{|V|} x_{ij}}, \tag{3}$$

where $|D_c|$ is the number of training documents belonging to the considered class c. To classify a new document with a given feature vector $d = (x_1, x_2, \cdots, x_{|V|})$, the multinomial naive Bayes classifier calculates a class specific probability for class c as

$$p(c|d) \propto p(c)p(d|\theta_c) = p(c) \frac{\left(\sum_{j=1}^{|V|} x_j\right)!}{\prod_{j=1}^{|V|}(x_j!)} \prod_{j=1}^{|V|} \theta_{cj}^{x_j}. \tag{4}$$

Here, $p(c)$ is the prior probability of class c which is estimated from a training set by $p(c) = |D_c|/|D|$ where $|D|$ is the total number of training documents in the used dataset. We estimate θ_{cj} in eq. (4) by using eq. (3) for each specified class c. The document is assigned to the class with highest probability $p(c|d)$. Taking the logarithm of eq. (4) and neglecting class-independent quantities, we obtain the decision function of the multinomial naive Bayes classifier:

$$w(c|d) = \log p(c) + \sum_{j=1}^{|V|} x_j \log \theta_{cj}. \quad (5)$$

The criterion is to assign d to the class c such that eq. (5) is maximized.

2.2 Poisson classifier

A well-known approach to obtaining high-performance generative probabilistic classifiers is to construct classifiers in a hierarchical manner by using conjugate prior/likelihood combinations. Studies following this approach have already been reported for the Dirichlet/multinomial [20], gamma/negative binomial [21], beta/binomial [22], and gamma/Poisson [23] combinations. We have reported that the beta/binomial and gamma/Poisson pairs give classification performance similar to that of support vector machines and clearly surpass that of multinomial naive Bayes classifier [23]. Here, however, we do not deal with such sophisticated hierarchical models and focus our attention toward building simpler classifiers which allow easy and effective implementation similarly to the multinomial naive Bayes classifier. For this reason, we do not employ the formulation of Kim et al. [14, 15] and instead use a simpler formulation that is basically the same as the formulation of Eyheramendy et al. [16] for our Poisson classifier.

Assumptions used to build the Poisson classifier are very similar to those of the multinomial naive Bayes:

1. An individual document is assumed to be represented as a vector of word counts.
2. The probability of the occurrence of a document d is a product of independent terms, each of which represents the probability of the number of emissions (i.e., the count) of an individual word.
3. The probability of the number of emissions is given by the usual Poisson distribution.

From the third assumption, the probability that there are x_{ij} occurrences of word t_j in the ith document belonging to class c is given by the usual Poisson distribution in the following form:

$$p(x_{ij}|c) = \frac{e^{-\lambda_{cj}} \lambda_{cj}^{x_{ij}}}{x_{ij}!}. \quad (6)$$

Here, λ_{cj} is the expected number of occurrences of t_j in a document belonging to class c and is estimated by

$$\hat{\lambda}_{cj} = \frac{C_1 + \sum_{i=1}^{|D_c|} x_{ij}}{C_2 + |D_c|}, \quad (7)$$

where C_1 and C_2 are smoothing parameters to prevent $\hat{\lambda}_{cj}$ from being zero, and $|D_c|$ the number of training documents belonging to class c. Note that the smoothing used in eq. (7) is similar to the Laplace smoothing used in eq. (3). Following [16], we set $C_1 = 0.001$ and $C_2 = 1$.

Combining the second assumption with eq. (6), the conditional probability of the occurrence of a document $d = (x_1, x_2, \cdots, x_{|V|})$ given class c is expressed as

$$p(d|c) = \prod_{j=1}^{|V|} \frac{\lambda_{cj}^{x_j} \exp(-\lambda_{cj})}{x_j!} \propto \prod_{j=1}^{|V|} \lambda_{cj}^{x_j} \exp(-\lambda_{cj}), \tag{8}$$

and thus a class specific probability for class c and the decision function, corresponding to eqs. (4) and (5) of the multinomial case, respectively, are given by

$$p(c|d) = p(c)p(d|c) = p(c) \prod_{j=1}^{|V|} \frac{\lambda_{cj}^{x_j} \exp(-\lambda_{cj})}{x_j!}, \tag{9}$$

$$w(c|d) = \log p(c) + \sum_{j=1}^{|V|} (x_j \log \lambda_{cj} - \lambda_{cj}), \tag{10}$$

for the Poisson classifier. In the training phase, the parameters of the Poisson distributions are evaluated through the estimator, eq. (7), for each possible class and then in the test phase, the classifier assigns the class c that has the highest value of the decision function, eq. (10), to a test document.

2.3 K-mixture classifier

For the K-mixture classifier, the third assumption of the Poisson classifier described above is replaced with the following assumption: "The probability of the number of emissions is given by the K-mixture distribution." The other two assumptions remain in their original forms. The new assumption leads us to the expression of the probability of x_{ij} occurrences of word t_j in the ith document belonging to class c as

$$p(x_{ij}|c) = (1 - \alpha_{cj})\delta_{x_{ij},0} + \frac{\alpha_{cj}}{\beta_{cj} + 1} \left(\frac{\beta_{cj}}{\beta_{cj} + 1} \right)^{x_{ij}}, \tag{11}$$

where α_{cj} and β_{cj} are parameters of the K-mixture distribution satisfying $0 < \alpha_{cj} < 1$ and $0 < \beta_{cj}$, respectively, and the $\delta_{x_{ij},0}$ is Kronecker's delta [1, 6]. Since we used the method of moments to estimate the parameters, the estimators of α_{cj} and β_{cj} are given by

$$\hat{\beta}_{cj} = \frac{1}{2} \left(\frac{\hat{\sigma}_{cj}^2}{\hat{\lambda}_{cj}} + \hat{\lambda}_{cj} - 1 \right), \tag{12}$$

$$\hat{\alpha}_{cj} = \frac{\hat{\lambda}_{cj}}{\beta_{cj}}, \tag{13}$$

where $\hat{\lambda}_{cj}$ is the smoothed sample mean given by eq. (7) and $\hat{\sigma}_{cj}^2$ is the sample variance defined as

$$\hat{\sigma}_{cj}^2 = \frac{1}{|D_c| - 1} \sum_{i=1}^{|D_c|} \left(x_{ij} - \hat{\lambda}_{cj}\right)^2. \tag{14}$$

Equations (12) and (13) can be derived by solving the expressions of mean and variance of the K-mixture given by Church and Gale [1] for α and β.

The second assumption with eq. (11) yields the conditional probability of document $d = (x_1, x_2, \cdots, x_{|V|})$ given class c in the following form:

$$p(d|c) = \prod_{j=1}^{|V|} p(x_j|c) = \prod_{j=1}^{|V|} \left\{ (1 - \alpha_{cj})\delta_{x_j,0} + \frac{\alpha_{cj}}{\beta_{cj} + 1} \left(\frac{\beta_{cj}}{\beta_{cj} + 1} \right)^{x_j} \right\}. \tag{15}$$

Thus we arrive at the decision function:

$$\begin{aligned} w(c|d) &= \log p(c) + \sum_{\{j|x_j=0\}} \log\left(1 - \alpha_{cj} + \frac{\alpha_{cj}}{\beta_{cj} + 1}\right) \\ &+ \sum_{\{j|x_j>0\}} \left\{ \log \alpha_{cj} - (1 + x_j) \log(\beta_{cj} + 1) + x_j \log \beta_{cj} \right\} \end{aligned} \tag{16}$$

The decision of the K-mixture classifier is to assign document d to class c such that eq. (16) is maximized.

2.4 Negative binomial classifier

For the negative binomial classifier, we replace the third assumption with the following statement: "The probability of the number of emissions is given by the negative binomial distribution." The probability of x_{ij} occurrences of word t_j in the ith document belonging to class c can be expressed as

$$P(x_{ij}|c) = \binom{N_{cj} + x_{ij} - 1}{x_{ij}} p_{cj}^{x_{ij}} (1 + p_{cj})^{-N_{cj} - x_{ij}}, \tag{17}$$

where $N_{cj} > 0$ and $p_{cj} > 0$ are parameters of the negative binomial distribution [1]. As in the K-mixture classifier, we used the method of moments to estimate the parameters N_{cj} and p_{cj}, which results in the estimators being expressed in the form:

$$\hat{p}_{cj} = \frac{\hat{\sigma}_{cj}^2}{\hat{\lambda}_{cj}} - 1, \tag{18}$$

$$\widehat{N}_{cj} = \frac{\hat{\lambda}_{cj}}{\hat{p}_{cj}}, \tag{19}$$

where $\hat{\lambda}_{cj}$ is the smoothed sample mean given by eq. (7) and $\hat{\sigma}_{cj}^2$ is the sample variance given by eq. (14). Here, Equations (18) and (19) are obtained by solving the expressions of mean and variance of the negative binomial given by [1] for the parameters.

The probability of the document d belonging to class c is thus calculated by

$$P(d|c) = \prod_{j=1}^{|V|} p(x_j|c) = \prod_{j=1}^{|V|} \left\{ \binom{N_{cj} + x_j - 1}{x_j} p_{cj}^{x_j} (1 + p_{cj})^{-N_{cj} - x_j} \right\}, \tag{20}$$

which is modified to give the decision function of the negative binomial classifier:

$$\begin{aligned} w(c|d) = \log p(c) \\ + \sum_{j=1}^{|V|} \{ \log \Gamma(N_{cj} + x_j) \\ - \log \Gamma(N_{cj}) + x_j \log p_{cj} - (N_{cj} + x_j) \log(1 + p_{cj}) \}. \end{aligned} \tag{21}$$

Note that we have substituted factorials with Gamma functions through the relation $\Gamma(n) = (n-1)!$ and have omitted the term $\log \Gamma(x_j + 1)$ that is independent of class label c and thus not necessary for classification purposes. The substitution of factorials with a gamma function is needed when x_j takes a real, non-integer value, which occurs through the procedures of document length normalization described in the next subsection. The decision of the negative binomial classifier is to assign d to the class c such that eq. (21) is maximized.

2.5 Normalization of document length

Thus far we have neglected the fact that the document lengths in the considered dataset differ from one another. In other words, we have assumed that each document in the dataset has the same length in terms of total word count. Of course, this is not necessarily true. Since the usual Poisson, K-mixture, and negative binomial distributions express the probability of a number of events occurring *in a fixed interval*, it is obvious that some normalization of document length is necessary when we try to apply these models to document classification. To normalize all the different lengths of training documents to be a predefined standard value, we used two different methods: L_1 normalization and pseudo-document normalization.

2.5.1 L_1 normalization

We consider the ith training document in class c: $d_{ci} = (x_{i1}, x_{i2}, \cdots, x_{i|V|})$. The document length of d_{ci} in an L_1 sense is simply given by the total number of occurrences of all terms:

$$l_i = \sum_{j=1}^{|V|} x_{ij}. \tag{22}$$

The normalization of the L_1 norm of document vector d_{ci} to be a predefined standard value of l_0, can be achieved through the conversion of each word count in d_{ci} by using

$$x_{0ij} = w_i x_{ij}, \tag{23}$$

where w_i is the ratio of the actual length l_i to the normalized length l_0; that is,

$$w_i = \frac{l_i}{l_0}. \tag{24}$$

To obtain the parameters of the usual Poisson, K-mixture, and negative binomial models for a normalized dataset in which each length of all the training documents is normalized to be exactly l_0, we use following procedure.

- The smoothed sample mean, eq. (7), is estimated from x_{0ij} given by eq. (23) instead of using the original count value, x_{ij}. We use the notation $\hat{\lambda}_{0cj}$ for the sample mean obtained in this manner, which expresses the sample mean of word occurrences of t_j over all the training documents in class c for the normalized dataset.

- The sample variance, eq. (14), is replaced with that using x_{0ij} and $\hat{\lambda}_{0cj}$, and the resultant variance is denoted as $\hat{\sigma}_{0cj}$, indicating the sample variance of word t_j in class c for the normalized dataset.

- The parameters of the K-mixture distribution for the normalized dataset, denoted by $\hat{\beta}_{0cj}$ and $\hat{\alpha}_{0cj}$, are estimated by eqs. (12) and (13), respectively, by changing $\hat{\lambda}_{cj}$ and $\hat{\sigma}_{cj}$ to $\hat{\lambda}_{0cj}$ and $\hat{\sigma}_{0cj}$.

- The parameters of the negative binomial distribution for the normalized dataset, denoted by $\hat{p}_{0cj}$ and $\hat{N}_{0cj}$, are estimated by eqs. (18) and (19), respectively, by changing $\hat{\lambda}_{cj}$ and $\hat{\sigma}_{cj}$ to $\hat{\lambda}_{0cj}$ and $\hat{\sigma}_{0cj}$.

The procedure for L_1 normalization described above is computationally simpler than the procedure for the pseudo-document normalization presented below.

2.5.2 Pseudo-document normalization

This normalization method is basically the same as proposed by [17] and [18]. In this method, all the training documents belonging to class c are firstly concatenated into a single huge document. The resultant length of this huge document is given by $L = \sum_{i=1}^{|D_c|} l_i$ where l_i is defined by eq. (22) and $|D_c|$ the number of training documents belonging to class c. Then, the huge document is split into equally sized pseudo-documents, each of which has exactly l_0

words. We then regard all the pseudo-documents obtained in this manner as the training set of normalized documents for class c and reconstruct the document vector d_{ci} by counting occurrences of each word in each of the pseudo-documents. Since each of the pseudo-documents has a predefined standard document length l_0, we denote the component of the reconstructed vector as x_{0ij}, which can be used in eqs. (7) and (14) to obtain the sample mean and the variance for normalized dataset without any corrections. ($|D_c|$ in eqs. (7) and (14), the number of training documents belonging to class c, should be reinterpreted as the number of pseudo-documents for this case.) Again, we denote the mean and variance as $\hat{\lambda}_{0ij}$ and $\hat{\sigma}_{0ij}$, respectively.

Estimating parameters of the K-mixture distribution for normalized dataset, $\hat{\beta}_{0cj}$ and $\hat{\alpha}_{0cj}$, and estimating those of the negative binomial distribution, $\hat{p}_{0cj}$ and $\hat{N}_{0cj}$, are also straightforward; explicitly, the estimation of these parameters can be achieved by using eqs. (12), (13), (18), and (19) directly with $\hat{\lambda}_{0ij}$ and $\hat{\sigma}_{0ij}$ obtained from the procedures described above.

2.5.3 Conversion of parameters for non-normalized test document

We consider the case where we try to classify a test document having an actual word count l. It has been shown that if we estimate the distribution parameters of the Poisson, K-mixture, and negative binomial distributions with a normalized dataset in which each document length is normalized to be exactly l_0, then the parameters for the test document having the actual length l should be given as follows [1] :

$$\hat{\lambda}_{cj} = w\,\hat{\lambda}_{0cj}, \quad \text{(Poisson)} \tag{25}$$

$$\hat{\alpha}_{cj} = \hat{\alpha}_{0cj}, \;\; \hat{\beta}_{cj} = w\,\hat{\beta}_{0cj}, \quad \text{(K-mixture)} \tag{26}$$

$$\hat{N}_{cj} = \hat{N}_{0cj}, \;\; \hat{p}_{cj} = w\,\hat{p}_{0cj}, \quad \text{(negative binomial)} \tag{27}$$

where the parameters with subscript 0 on the right-hand side are those estimated for the normalized dataset obtained through the L_1 normalization or the pseudo-document normalization, and the parameters without subscript 0 on the left-hand side are those for the test document having the actual length l. In these equations, w is the ratio of the actual length to the normalized length; that is, $w = \frac{l}{l_0}$.

In the training phase of each classifier, we used one of the two normalization methods described above to obtain the parameters for the normalized dataset, and then in the test phase, eqs. (25)~(27) were used to adjust the parameters to the values suitable for the non-normalized test document.

3 EXPERIMENTAL EVALUATION

To clarify the characteristics of the proposed three classifiers with the usual Poisson, K-mixture, and negative binomial models, we performed text classification experiments using three standard document corpora. In the experiments, the performance of the proposed three classifiers is compared with that of the baseline multinomial naive Bayes classifier.

3.1 Dataset

In our experiments, we chose three different datasets that represent a wide spectrum of text classification tasks.

The first one is the 20 Newsgroups dataset which was originally collected with a netnews-filtering system [24] and contains approximately 20,000 documents that are partitioned nearly evenly across 20 different UseNet newsgroups. We use the 20news-18828 version from which cross-posts have been removed to give a total of 18,828 documents. (Original dataset is available from: http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.data.html. 20News-18828 is available from: http://people.csail.mit.edu/jrennie/20Newsgroups/.) Consequently, 20 Newsgroups is a single-labeled dataset with approximately even class distribution, and the task is to apply one of the 20 possible labels to each test document. We build an initial vocabulary from all words left after stop word, punctuation, and number token removal. Uppercase letters are converted to lowercase letters and no stemming algorithm is applied. Here, words are defined as alphabetical strings enclosed by whitespace. The size of the initial vocabulary is 110,492 words.

The second dataset is SpamAssassin which is available as part of the open-source Apache SpamAssassin Project 2 for public use. (The corpus is available online at http://spamassassin.apache.org/publiccorpus/.) It consists of email divided into three categories: “Easy Ham”, which is email unambiguously ham (i.e., not spam), “Hard Ham” which is not spam but shares many features with spam, and finally “Spam”. The task is to apply these three labels to test emails. We use the latest version of all datasets, and combine “easy ham” and “easy ham 2” datasets to form our Easy Ham dataset; similarly, “spam” and “spam 2” datasets are combined to form our Spam dataset. The preprocessing before building the initial vocabulary was the same as for the 20 Newsgroups. The resulting corpus is just over 6,000 messages with an initial vocabulary of 151,126 words.

The third test collection is the Industry Sector dataset which is a collection of corporate Web pages organized into hierarchical categories based on what a company produces or does. Although it has a hierarchy with three levels of depth, we do not take the hierarchy into account and use a flattened version of the dataset. This dataset contains a total of 9,555 documents divided into 104 categories. (We obtained the dataset from http://www.cs.umass.edu/ mccallum/code-data.html. Because it was found that one of the

original 105 categories was empty, the remaining 104 categories having documents were used in our experiments.) We use all 9,555 documents in our experiments without removing the multi-labeled documents because the fraction of multi-labeled documents is very small and the effect of these documents is negligible. (Only 15 documents out of 9,555 belong to two classes; thus, they cannot affect our results considerably.) The largest and smallest categories have 105 and 27 documents, respectively, and the average number of documents per category is 91.9. For this dataset, we remove HTML tags by skipping all characters between "<" and ">", and we did not use a stop list. The resulting vocabulary has 64,202 words.

For all three datasets, we use 10-fold cross-validation to make maximal use of the data. Ten obtained values of performance are averaged to give the final result.

3.2 Feature selection

To investigate the effect of vocabulary size on classification performance, we use a simple feature selection method based on the collection term frequency as follows. First, we count the collection term frequency, CF, which is the total frequency of each word throughout the entire dataset. Then, we select all words that satisfy $CF \geq N_0$ where N_0 is a predefined integer. The feature selection by CF is one of the simplest methods, but is sufficient for the task at hand, namely, comparing different classifiers at each vocabulary size. The resultant vocabulary sizes after feature selection are summarized in Table 1.

Table 1: Vocabulary size obtained by feature selection with CF.

Feature selection	20 Newsgroups	SpamAssassin	Industry Sector
Initial vocabulary	110,492	151,126	64,202
$CF \geq 2$	64,065	53,886	37,634
$CF \geq 5$	34,124	21,258	21,216
$CF \geq 10$	21,697	12,749	14,317
$CF \geq 20$	13,709	7,754	9,455
$CF \geq 50$	7,314	3,869	5,329
$CF \geq 100$	4,252	2,085	3,233
$CF \geq 200$	2,180	1,077	1,770
$CF \geq 500$	748	402	665
$CF \geq 1000$	255	176	290

Count-valued document vectors $\{d_{ci}\}$ are constructed from document term frequency (number of occurrences of a considered word in a document) for each word in a vocabulary at each vocabulary level. Since we use the 10-fold cross-validation, 1/10 of the original count vectors $\{d_{ci}\}$ are used as test vectors and the rest are used as original training vectors. In the training phase, the original training vectors are supplied to the three classifiers which normalize the training vectors by L_1 normalization or pseudo-document normalization and then estimate

the distribution parameters. In addition to the two normalization methods, the distribution parameters without any normalization are directly calculated from the original training vectors to clarify the effect of document length normalization. To classify test vectors in the test phase, the classifiers use the three types of distribution parameters: those obtained without normalization, those obtained by L_1 normalization and those obtained by pseudo-document normalization.

3.3 Implementation issues

All the classifiers used in this study are implemented in the Java programming language. Supplementary information is as follows:

- For calculating the sample variance, we slightly modified eq. (14) for the following reason. The estimator of p_{cj}, eq. (18), requires $\hat{\sigma}_{cj}^2 > \hat{\lambda}_{cj}$ to satisfy the constraint $\hat{p}_{cj} > 0$. To ensure that the constraint is satisfied, if $\hat{\sigma}_{cj}^2$ calculated by eq. (14) is less than or equal to $\hat{\lambda}_{cj}$, we always replace the original value of $\hat{\sigma}_{cj}^2$ with $\hat{\sigma}_{cj}^2 = \hat{\lambda}_{cj} + \varepsilon$ in which a constant ε is set to 0.1 after a preliminary classification experiment on the 20 Newsgroups dataset. This happens when a considered word t_j fails to appear in any of the training vectors for the considered class. In this case, $\hat{\sigma}_{cj}^2$ calculated by eq. (14) is approximately equal to $\hat{\lambda}_{cj}^2$ and thus much smaller than $\hat{\lambda}_{cj}$.

- In L_1 normalization, we use $l_0 = 1{,}000$ for the normalized document length while in the case of the pseudo-document normalization, we set $l_0 = 100$. The value of $l_0 = 1{,}000$ for L_1 normalization was determined after a preliminary experiment on 20 Newsgroups dataset (we tried $l_0 = 100, 1000, 10000$ and found that $l_0 = 1000$ gives the best classification performance.) , while $l_0 = 100$ for pseudo-document normalization was determined to ensure a sufficient number of pseudo-documents for all categories in the three datasets used.

- To compute the log of the gamma function in eq. (21), components available in the Apache Commons Mathematics Library (http://commons.apache.org/math/) are used.

4 RESULTS

As in our previous study [23], we also use the simplest measure of classification performance in this study, that is, accuracy, which is simply defined as the ratio of the total number of correct decisions to the total number of test documents in the dataset used. Note that for a single-labeled dataset and a single-labeled classification scheme as in this work, the micro-averaged precision and recall are equivalent, and hence equal to the F1 measure [25], which we call "accuracy" here.

4.1 Effect of document normalization

Figures 1, 2, and 3 show the performance of the classifiers for the 20 Newsgroups, SpamAssassin, and Industry Sector datasets, respectively.

In all these figures, the top-left plot (a) shows classification accuracy without normalization, the middle-left plot (b) shows classification accuracy with L_1 normalization, and the bottom-left plot (c) shows classification accuracy with pseudo-document normalization. The plots on the right side in Figs. 1, 2, and 3 (i.e., plots (a'), (b'), and (c')), show the same information as plots (a), (b), and (c), respectively, but with the horizontal axes on a logarithmic scale to show the lower vocabulary region clearly. Note that the accuracies of the multinomial classifier in each figure are identical in all plots (a)~(c'), because L_1 or pseudo-document normalization was only applied to the classifiers using the Poisson, negative binomial, and K-mixture models and was not applied to the multinomial naive Bayes classifier. In Figs. 1, 2, and 3, the accuracy curves of the multinomial classifier in plots (b) and (c), and those in plots (b') and (c') are thus simple replicas of those in plot (a) and plot (a'), respectively.

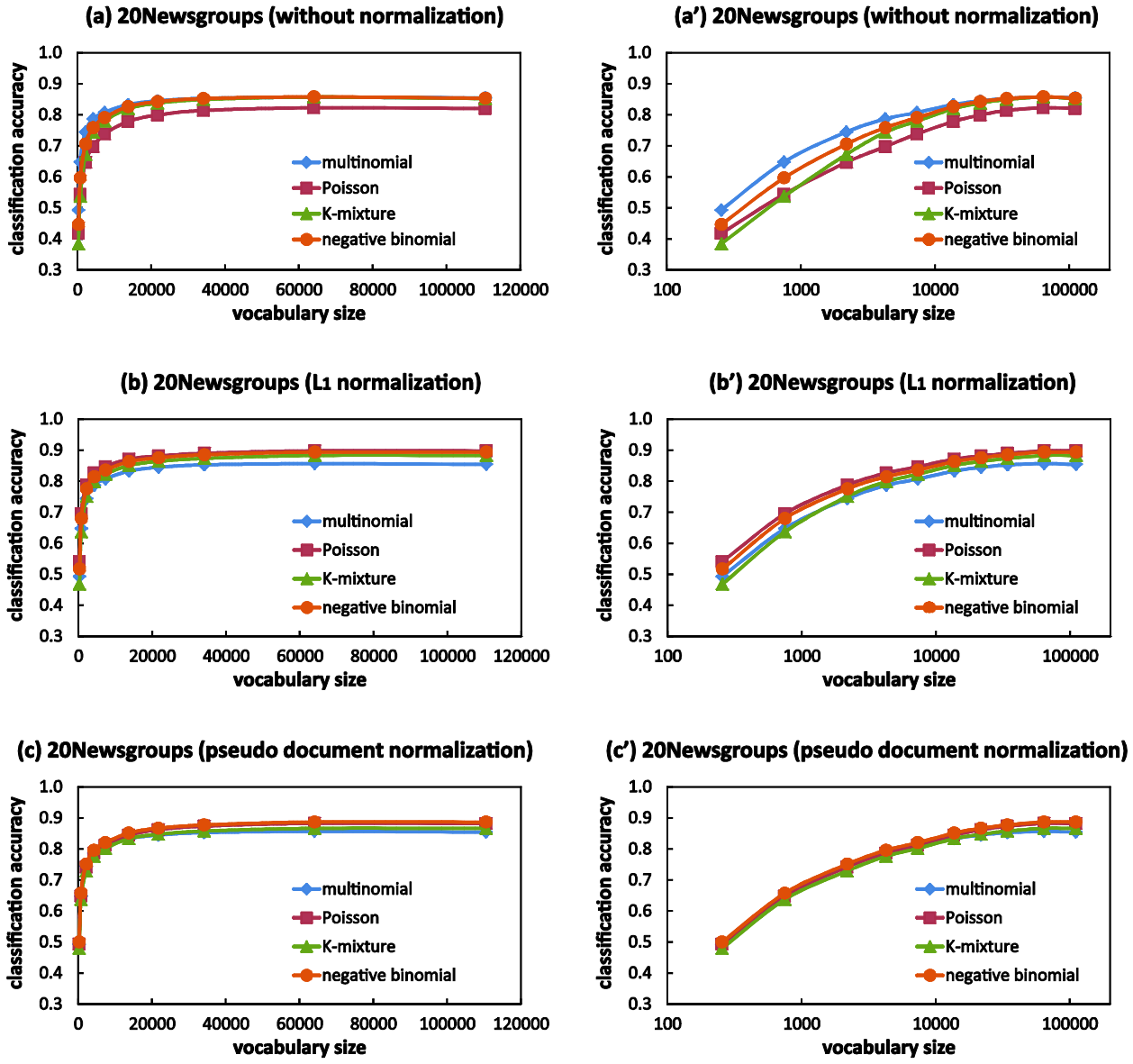


Figure 1: Classification performance of examined four classifiers on 20 Newsgroups dataset. (a) and (a') show the performance without document length normalization; (b) and (b'), with L_1 normalization; and (c) and (c'),

pseudo-document normalization. In (a), (b), and (c), the horizontal axes are linear while they are logarithmic in (a'), (b') and (c') in order to show the lower vocabulary region clearly.

The reason for this special treatment of the multinomial naive Bayes classifier is that the estimated parameter $\hat{\theta}_{cj}$ with eq. (3) for this classifier represents the probability of selecting the word t_j at an arbitrary position of documents in class c *with any arbitrary document length*. Similarly, the probability of the document d_{ci} for the multinomial naive Bayes calculated by use of eq. (2) is valid for documents in class c with any arbitrary document length $\sum_{j=1}^{|V|} x_{ij}$. On the other hand, when we calculate the probability of the document d for the Poisson, K-mixture and negative binomial models by use of eqs. (8), (15) and (20), the document length of d should be normalized to l_0, which is the document length used at estimating parameters. This is because the Poisson distribution describes the probability of count data occurring *in a fixed interval*, and as a consequence, the probability of x_{ij} occurrence of word t_j given by use of eqs. (6), (11) and (17) for the Poisson, K-mixture and negative binomial models are, in a rigorous sense, only valid for the documents with fixed length l_0.

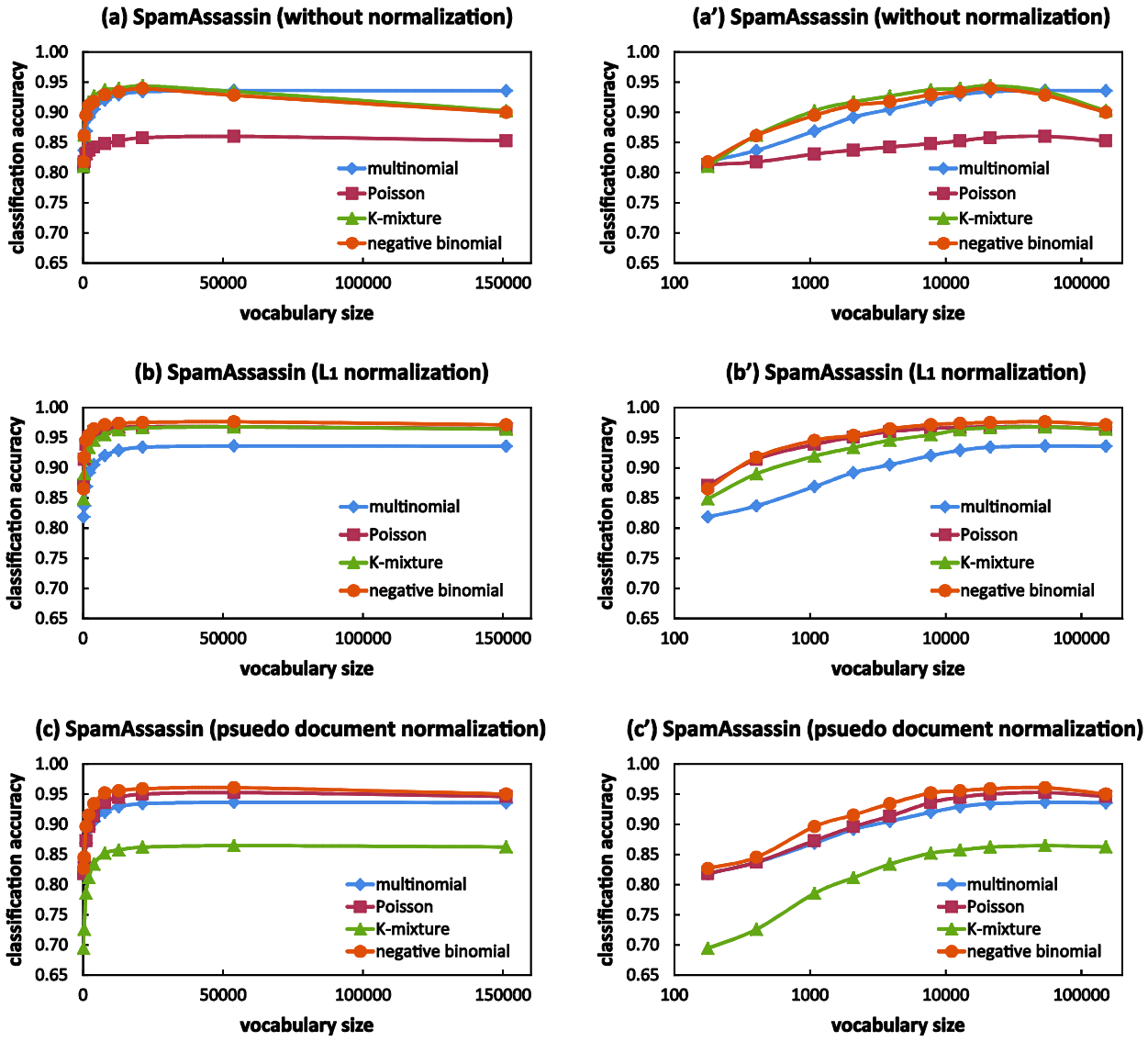


Figure 2: The same as in Fig. 1, but for the SpamAssassin dataset.

Based on the results shown in Figs. 1, 2, and 3, we first consider the effect of document length normalization on classification accuracy. The overall trends of the accuracy curves in these figures clearly indicate that the normalization of document length is fundamentally important to achieve better performance for the classifiers using the Poisson, negative binomial, and K-mixture models. Detailed observations are given below.

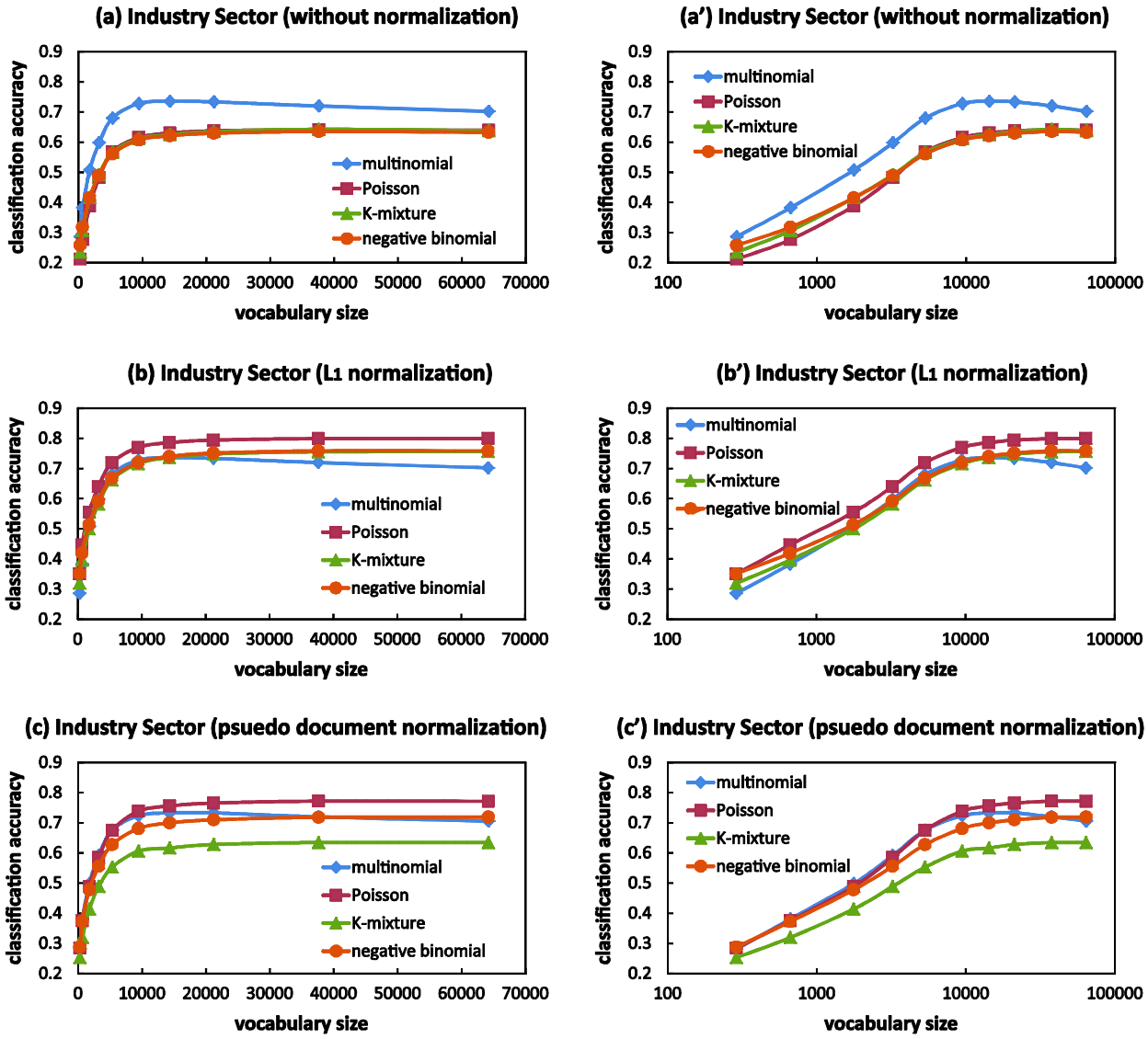


Figure 3: The same as in Fig. 1, but for the Industry Sector dataset.

- For the 20 Newsgroups dataset, it can be seen from Figs. 1(a) and 1(a') that the performance of the K-mixture and negative binomial classifiers without normalization are similar to that of the baseline multinomial classifier especially in the higher vocabulary region and that the Poisson classifier is apparently worse than that of the multinomial classifier for the non-normalized data. On the other hand, Figs. 1(b), (b'), (c), and (c') show that the accuracies of the Poisson, K-mixture, and negative binomial classifiers are higher than that of the multinomial classifier for normalized data.

- For the SpamAssassin dataset, the results are consistently the same or very close to those of the 20 Newsgroups dataset. Figures 2(a) and (a') show that the K-mixture and negative binomial classifiers achieve accuracy similar to that of the multinomial classifier but the Poisson classifier fails to achieve that level of performance for non-normalized data. It is also confirmed from Figs. 2(b), (b'), (c), and (c') that both types of normalization bring better performances to the classifiers using the Poisson, negative binomial, and K-

mixture models than the one using the baseline multinomial model. The K-mixture classifier with pseudo-document normalization is the only exception and performs worse than the multinomial classifier does (Figs. 2(c) and (c')).

- The influence of document length normalization on classification accuracy is most evident for the Industry Sector dataset as seen in Fig. 3. Compared with the multinomial classifier, the Poisson, negative binomial, and K-mixture classifiers appear to give worse performance for non-normalized data (Figs. 3(a) and (a')), while they perform better than the multinomial classifier does for normalized data especially in the higher vocabulary region (Figs. 3(b), (b'), (c), and (c')). Again, the K-mixture classifier with pseudo-document normalization (Figs. 3 (c) and (c')) is the only exception and exhibits much worse performance than the multinomial classifier.

In short, Figs. 1~3 show that the document length normalization is effective for improving the performance with the Poisson, negative binomial, and K-mixture models. Although the degree of improvement differs by dataset, normalization typically achieves much better performance than that of the baseline multinomial classifier, in contrast to the case that non-normalized data are used.

In a comparison between the two different types of normalization, L_1 normalization seems to bring about better improvement than pseudo-document normalization does, and this tendency is clearest for the K-mixture classifier, as seen in plots (b') and (c') of Figs. 2 and 3. The difference in the degree of improvement between the two normalization methods can be ascribed to the following reason. In the pseudo-document normalization, each term occurrence is treated as being equally important while in the L_1 normalization, the event of a word occurrence has remarkably different weight according to the original document length. This is because, in the L_1 normalization, the occurrence of a word in a short document more heavily weighted than the occurrence of the same word in a long document. The conversion of the L_1 normalization in this manner is reasonable and considered to bring about the better performance because, compared with long documents, short documents usually have fewer unnecessary terms that are irrelevant to the topic and the ratio of informative terms that represent a concept of the topic is higher.

4.2 Comparative performance behavior

In this subsection, we compare the four classifiers in term of classification performance. First, we consider the performance in the non-normalized case. Clearly, the multinomial classifier is the best performer when we use non-normalized data, as is clearly exhibited in plot (a') of Figs. 1, 2, and 3. As for the other three classifiers in the non-normalized case, the negative binomial and K-mixture classifiers give similar performance and are superior to the Poisson classifier. Indeed, the negative binomial and K-mixture classifiers perform similarly to the multinomial classifier and apparently better than the Poisson classifier (Figs. 1(a), 1(a'), 2(a), and 2(a')). The superiority of the negative binomial and K-mixture classifiers over the Poisson classifier for non-normalized data can be attributed to their flexibility in modeling text because

they can describe the overdispersion of word frequency which is often encountered in real texts but is not modeled well by the Poisson distribution.

We next consider the cases where normalized data are used. The overall trends in the accuracy curves suggest that the Poisson, negative binomial, and K-mixture models achieve much better performance than the multinomial model for normalized data, as described in the previous subsection. To examine which model is best for normalized data, we further compare the three classifiers except the multinomial classifier on the basis of the results shown in Figs. 1∼3. Observation of the six cases (3 datasets × two normalization methods) shows that the usual Poisson classifier performs best in two of six cases (Figs. 3(b') and 3(c')), the negative binomial performs best in one case (Fig 2(c')), and they perform similarly in the other three cases (Figs. 1(b'), 1(c'), and 2(b')). Therefore, the usual Poisson classifier appears to be the best performer, the negative binomial classifier is found to perform at a similarly high level, and compared with the K-mixture classifier, they are better for normalized data. These results raise the question as to why the usual Poisson model, which performs poorly for non-normalized data as described above, performs well for normalized data. In the next section, this question is discussed by considering the trade-off between fit and model complexity.

5 DISCUSSION

To examine the behavior of the usual Poisson, negative binomial, and K-mixture models for normalized datasets from a different perspective, we attempt to calculate the two most commonly used penalized model selection criteria, the Akaike's information criterion (AIC) and the Bayesian information criterion (BIC), for the three datasets and we investigate the relation between these criteria and classification performance. In general, penalized model selection criteria are statistics of the form [26, 27]

$$-2L(\hat{\theta}_M) + kp_M, \tag{28}$$

where $\hat{\theta}_M$ is a parameters vector obtained by maximum likelihood estimation (MLE), $L(\hat{\theta}_M)$ is the log-likelihood, k is a known positive quantity, and p_M is the number of parameters in the model under consideration. The maximized log-likelihood obtained through MLE, that is, $L(\hat{\theta}_M)$ in the first term of eq. (28), reflects the fit of the considered model to the observed data, while p_M in the second term is regarded as a measure of the complexity of a considered model. The second term penalizes models for the number of parameters used. The two terms of eq. (28) thus pull in opposite directions, apparently expressing a trade-off between fit and model complexity. The penalized model selection criteria are intended to help select the best model from among several competing models, that is, a value of a criterion is calculated for each model under consideration, and the model with the smallest value is chosen as the best one.

The two most commonly used penalized model selection criteria, the AIC and BIC, are defined as [27-29]

$$AIC = -2L(\hat{\theta}_M) + 2p_M, \tag{29}$$

$$BIC = -2L(\hat{\theta}_M) + p_M \log n, \tag{30}$$

where n is the number of observations. The difference between their penalty terms seen in eqs. (29) and (30) arises from their different foundations [27]. In the following, we calculate AIC and BIC for each word t_j for each class c in the considered dataset by using

$$AIC_M(cj) = -2\sum_{i=1}^{|D_c|} \log p_M(x_{ij}|c) + 2p_M, \tag{31}$$

$$BIC_M(cj) = -2\sum_{i=1}^{|D_c|} \log p_M(x_{ij}|c) + p_M \log|D_c| \tag{32}$$

where $|D_c|$ is the number of documents belonging to considered class c, and the subscript M specifies the model and indicates Poisson, negative binomial, or K-mixture. In eqs. (31) and (32), $p_M(x_{ij}|c)$ means the model-dependent probability of x_{ij} occurrences of word t_j in the ith document of class c, and is given by eq. (6) for the Poisson, by eq. (11) for the K-mixture, and by eq. (17) for the negative binomial, respectively. Note that the parameters for each distribution model (i.e., $\hat{\lambda}_{cj}$ for the Poisson, $\hat{\alpha}_{cj}$ and $\hat{\beta}_{cj}$ for the K-mixture, and $\hat{N}_{cj}$ and $\hat{p}_{cj}$ for the negative binomial) are estimated from all the documents belonging to class c in the considered dataset and are used to calculate $p_M(x_{ij}|c)$ for each model. We used the method of moments described in the previous section to estimate distribution parameters. (Parameters obtained by the method of moments do not coincide with those obtained by MLE in a strict sense. However, our experiences showed that an iterative calculation to obtain the MLE solution for the negative binomial parameters, which is given by [1], is not stable and can be easily affected by outliers. A similar trend was observed for MLE of the K-mixture parameters, and thus, we used the method of moments which offers more robust estimations.) Also, at the estimation of parameters and at the calculation of $p_M(x_{ij}|c)$, all the documents are normalized by L_1 normalization with $l_0 = 1000$. The number of parameters for each model, p_M, is set to be $p_P = 1$ (Poisson), $p_K = 2$ (K-mixture) and $p_{NB} = 2$ (negative binomial). To compare the AIC and BIC values with the classification performances, a further step of averaging AIC and BIC over all categories is needed because the classification performance was obtained from entire documents of the considered dataset and thus reflect averaged classification accuracy over all categories. The averaged AIC and BIC for each word t_j in the vocabulary are obtained through

$$AIC(\text{Poisson / K} - \text{mixture / negative binomial}, t_j) = \frac{1}{|C|}\sum_{c=1}^{|C|} AIC_M(cj), \tag{33}$$

$$BIC\big(\text{Poisson / K} - \text{mixture / negative binomial}, t_j\big) = \frac{1}{|C|}\sum_{c=1}^{|C|} AIC_M(cj), \tag{34}$$

where $|C|$ is the number of classes in the considered dataset, and $AIC_M(cj)$ and $BIC_M(cj)$ are the AIC and BIC of word t_j for class c as given by eqs. (31) and (32), respectively.

Figures 4, 5, and 6 show the scatter plots between two of three text models in terms of AIC and BIC for the 20 Newsgroups dataset, SpamAssassin dataset, and Industry Sector dataset, respectively. One data point in each plot of these figures corresponds to a word in the vocabulary of the considered dataset; we calculated the AIC or BIC for two different models by using eq. (33) or (34) and these values were used as the x- and y-coordinates of the data point.

From overall trends in AIC and BIC depicted in Figs. 4, 5, and 6, we can find that an arbitrary pair among the three models which has a strong positive correlation with each other in terms of AIC and BIC. Another finding is that AIC and BIC behave fundamentally the same. This can be explained from eqs. (31) and (32) by noting that the number of parameters, p_P, p_K, and p_{NB}, and the number of documents, $|D_c|$, are common for all words in a given class of a dataset under consideration, and hence the difference between AIC and BIC is always a common constant for all words.

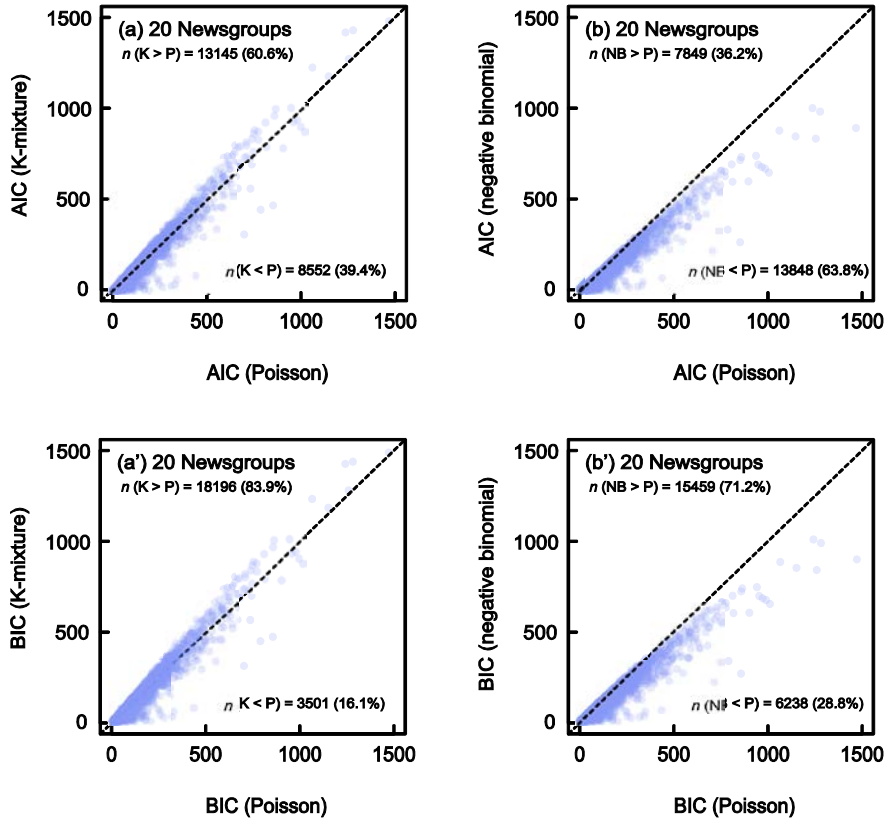


Figure 4: Scatter plots between two of three text models in terms of AIC and BIC for the 20 Newsgroups dataset. (a) shows the correlation between the Poisson model and the K-mixture model in terms of AIC, and (b)

shows that between the Poisson model and the negative binomial model. (a') and (b') are the same as (a) and (b), respectively, but show BIC. In all cases, the vocabulary level used is $CF > 10$ (21,697 words).

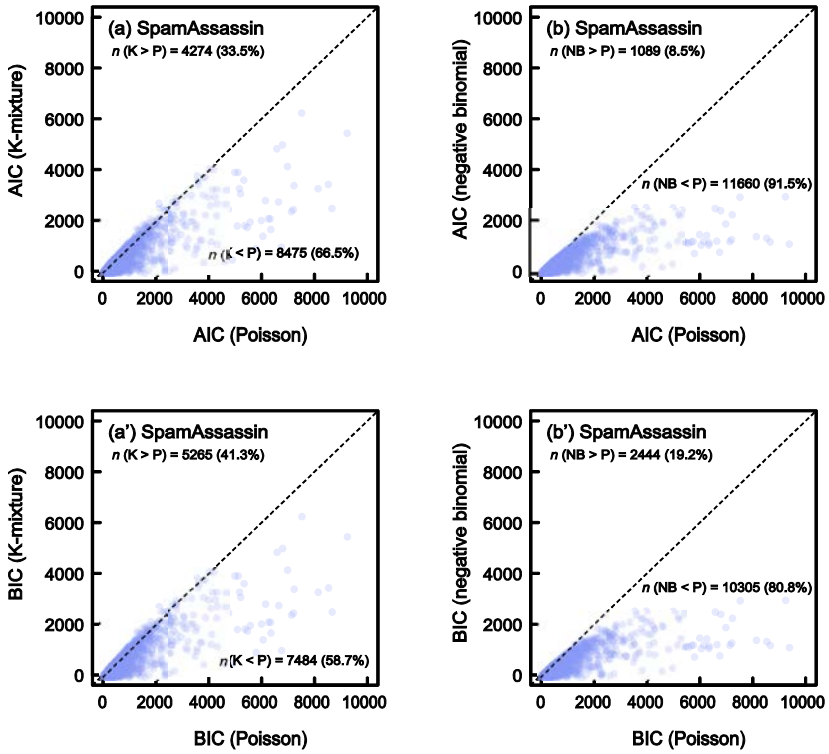


Figure 5: The same as in Fig. 4, but for the SpamAssassin dataset. In all cases (a)~(b'), the vocabulary level used is $CF > 10$ (12,749 words).

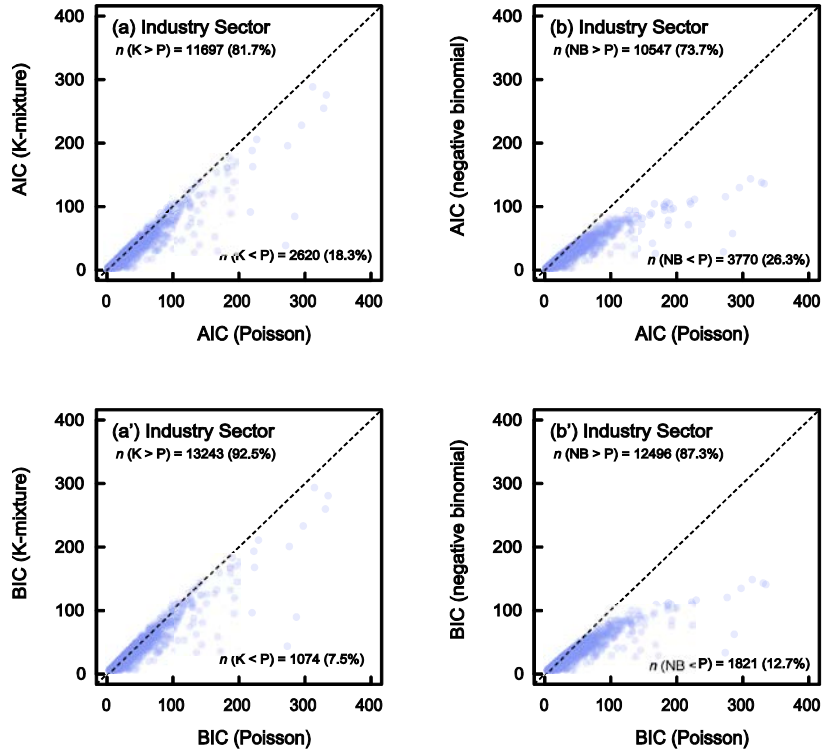


Figure 6: The same as in Fig. 4, but for the Industry Sector dataset. In all cases (a)~(b'), the vocabulary level used is $CF > 10$ (14,317 words).

Dotted diagonal lines in each plot of Figs. 4, 5, and 6 represent the function $y = x$ and the numbers of data points satisfying $y > x$ and $y < x$ are depicted in each plot. For example, in Fig. 4(b'), the x and y axes represent the BIC of the Poisson model and that of the negative binomial model, respectively, and the number of words satisfying $BIC(\text{negative binomial}) > BIC(\text{Poisson})$ located in the upper half-plane over the dotted line in the plot area is 'n(NB>P) = 15459 (71.2%)' and the number of data points satisfying $BIC(\text{negative binomial}) < BIC(\text{Poisson})$ located in the lower half-plane is 'n(NB < P) = 6238 (28.8%)'. The total of these two data points, 21697, gives the vocabulary of the 20 Newsgroups dataset chosen under the condition that $CF > 10$, as Table 1 shows. At first glance, the number of data points in the upper half-plane in Fig. 4(b') seems to be much smaller than that in the lower half-plane; however, a closer look indicates that the data points located in the upper half-plane are much more densely plotted and hence there are more points in the upper half-plane.

To compare the three models in terms of AIC and BIC, we tentatively use the numbers of data points in the upper and the lower half-planes. For example, from Fig. 4(b'), we find that the relation $BIC(\text{negative binomial}) > BIC(\text{Poisson})$ holds for about 70% words in the vocabulary and therefore we can conclude that the Poisson model is more suitable than the negative binomial for the 20 Newsgroups dataset because the former gives smaller BICs for most words. We have made similar comparisons for each scatter plot in Figs. 4, 5, and 6 and have found that the comparisons in terms of BICs are qualitatively consistent with the classification accuracy described in the previous section. Table 2 summarizes our comparisons of three text models in terms of BIC. In the statements of comparison of text models in Table 2, for example, 'NB > K-mixture≳ Poisson' for the SpamAssassin dataset, NB means negative binomial and the symbols '>' and '≳' should be read as "better than" and "better than or equivalent to", respectively. We have tentatively used the following evaluation criteria to compare the models. For comparing the K-mixture and Poisson models, if the percentage of n(K > P) is larger than 60% we conclude that 'Poisson > K-mixture', or if the percentage is less than 40% we conclude 'K-mixture > Poisson', and otherwise (the percentage is in the range of 40%~60%) we conclude that 'Poisson ≳ K-mixture' or 'K-mixture ≳ Poisson'. The same evaluation criteria have been used for comparing the negative binomial and Poisson models. For comparing the K-mixture and negative binomial models, if the difference in the ratio between n(K > P) and n(NB > P) is more than 10%, then we conclude that one is better than the other; otherwise we conclude that one is better than or equivalent to the other. The rather loose criteria with a 20% range of tolerance described above arise from a consideration that the absolute values of the ratios n(K > P) and n(NB > P) have a degree of uncertainty. The existence of this uncertainty is intuitively recognized from the fact that the ratio of, for example, n(K>P), calculated with AIC and that with BIC take different values by amount up to 20%. Since we do not have enough evidence to prove that either the AIC or BIC is better than

the other, the ratio of n(K>P) should be considered to have the same degree of uncertainty. The comparisons of three text models are derived from the following considerations:

- For 20 Newsgroups, n(K > P)=83.9% indicates that the Poisson model is better than the K-mixture model, n(NB > P)=71.9% indicates that the Poisson model is also better than the negative binomial model, and the comparison of n(K > P)=83.9% and n(NB > P)=71.9% leads us to conclude that the negative binomial model is better than the K-mixture model. Thus the results are summarized as ‘Poisson > NB > K-mixture’ as shown in Table 2.

- For SpamAssassin, n(K < P)=58.7% indicates that the K-mixture model is better than or equivalent to the Poisson model, n(NB < P)=80.8% shows that the negative binomial model is better than the Poisson model. Thus the results are summarized as ‘NB > K-mixture ≳Poisson’.

- For Industry Sector, n(K > P)=92.5% indicates that the Poisson model is better than the K-mixture model, n(NB > P)=87.3% also means that the Poisson model is better than the negative binomial model, and the comparison of n(K > P)=92.5% and n(NB > P)=87.3% leads us to conclude that the negative binomial model is better than or equivalent to the K-mixture model. The results are summarized as ‘Poisson > NB ≳ K-mixture’.

Table 2: The percentages of words in the upper and lower half-planes in the scatter plots of BICs (plot (a') and (b') in Figs. 4, 5, and 6) and comparison of text models derived from the numbers of the data points.

dataset	K-mixture vs. Poisson		NB vs. Poisson		comparison of text models
	n(K > P)	n(K < P)	n(NB > P)	n(NB < P)	
20 Newsgroups	83.9%	16.1%	71.2%	28.8%	Poisson > NB > K-mixture
Spam Assassin	41.3%	58.7%	19.2%	80.8%	NB > K-mixture ≳ Poisson
Industry Sector	92.5%	7.5%	87.3%	12.7%	Poisson > NB ≳ K-mixture

Table 3 shows the classification performance corresponding to Table 2 and the comparisons of models in terms of classification accuracy. As seen in Tables 2 and 3, the result of comparing the three models in terms of BIC (Table 2) and that in terms of classification accuracy (Table 3) are reasonably consistent with each other, and the slight discrepancy is only that ‘K-mixture ≳ Poisson’ for the SpamAssassin dataset in Table 2 is replaced with ‘Poisson ≳ K-mixture’ in Table 3. Of course, we can consider that this discrepancy is not a fundamental difference because there is some uncertainty in n(K>P) and therefore the comparison results in Tables 2 and 3 are fundamentally the same.

Table 3: Comparisons of the three models in terms of classification accuracy. For each dataset, the vocabulary used is determined by the condition *CF*>10 and all the document vectors are normalized to l_0=1000 by L_1 normalization. Values are shown as accuracy±σ where σ is the standard deviation calculated through 10-fold cross-validation.

dataset	classification accuracy			comparison of text models
	Poisson	K-mixture	NB	
20 Newsgroups	88.11±0.61%	86.38±0.58%	87.51±0.62%	Poisson > NB > K-mixture
Spam Assassin	96.70±0.84%	96.33±0.63%	97.38±0.76%	NB > Poisson ≳K-mixture
Industry Sector	78.58±1.43%	73.63±1.30%	73.98±1.37%	Poisson > NB ≳ K-mixture

We now consider the meaning of the consistency between Tables 2 and 3 described above. By definition, an information criterion such as AIC and BIC having the form of eq. (28) indicates that given a finite quantity of data available for modeling, a model with a higher degree of freedom will have greater instability, resulting in reduced prediction ability [27]. The situation is very similar in classification tasks [30, chapter 4]. If we try to build a classification model that fits the training data too well in order to lower the training error, then the generalization error in classifying unknown test data becomes larger due to overfitting. In this sense, the log-likelihood in eq. (28) corresponds to the degree of fitting in the training phase of classification and that represents how well the classification model fits the training data, and the second term of eq. (28) corresponds to the penalty for overfitting that will lead to misclassification in the test phase. The consistency between Tables 2 and 3 allows for an intuitive interpretation that both the degree of positive influence due to maximizing the log-likelihood with a precise description of the word distribution estimated through BIC and the degree of negative influence due to overfitting also estimated through BIC agree well with the actual trends, i.e., the actual amelioration and the deterioration in classification accuracy for the three models used.

We next consider the reason why the usual Poisson model performs better for a normalized dataset than the K-mixture and negative binomial models do while it behaves worst among the three models for a non-normalized dataset. Figures 7, 8, and 9 show the scatter plots between the mean and variance for each word in a vocabulary for the 20 Newsgroups, SpamAssassin, and Industry Sector datasets, respectively. The plots labeled (a) in these figures show the case for non-normalized datasets and the plots labeled (b) show for the normalized cases using L_1 normalization with $l_0 = 1000$. Comparing with plots (a) and (b), the positive correlation between the mean and variance appears to become stronger in the case of the normalized datasets compared with non-normalized datasets. In other words, when we want to describe the distribution of each word precisely, the mean and variance of each word are necessary in the case of a non-normalized dataset, while specifying both of these two values seems excessive in the case of a normalized dataset because these two values have strong positive correlation as seen in the plots labeled (b). This finding from Figs. 7, 8, and 9 can be

reinterpreted in terms of the number of model parameters needed to describe the word distribution because the mean and variance can be directly converted to $\hat{\alpha}_{cj}$ and $\hat{\beta}_{cj}$ for the K-mixture model and to $\hat{p}_{cj}$and $\widehat{N}_{cj}$ for the negative binomial model. From the finding described above, we can deduce a possible explanation of the good performance of the usual Poisson models for normalized datasets as follows. For non-normalized data, specifying two parameters in the distribution model is necessary to obtain a large value of log-likelihood in eq. (28) and thus the models having two parameters, (i.e., the K-mixture and negative binomial models), are advantageous over the Poisson model for non-normalized data. On the other hand, using two parameters to describe the word distribution in normalized datasets is expensive in the sense that the effect of decreasing the information criteria by maximizing the log-likelihood with two parameters is restrictive and thus the influence of the penalty term becomes larger for normalized data. This situation makes the one-parameter Poisson model superior to the two-parameter K-mixture and negative binomial models in the case of a normalized dataset.

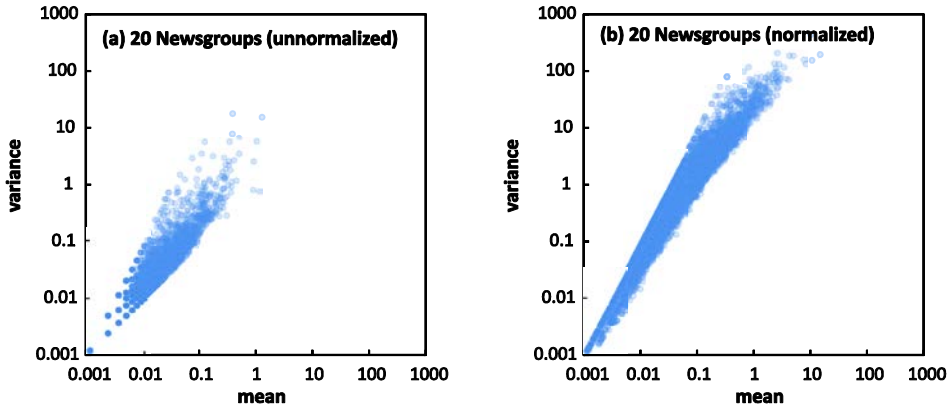


Figure 7: Scatter plots between the mean and variance of each word in a vocabulary for the 20 Newsgroups dataset. (a) shows the case of non-normalized data and (b) shows the case of normalized data by use of L_1 normalization with l_0=1000. The vocabulary level used is *CF*>10 (21,697 words) and the means and variances are calculated from all the documents belonging to the 'alt.atheism' category (the first category in alphabetical order) from the dataset.

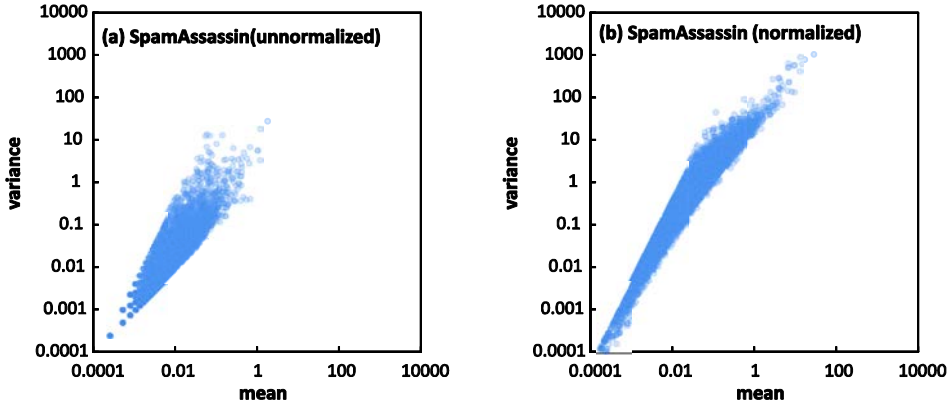


Figure 8: The same as in Fig. 7, but for the SpamAssassin dataset. The vocabulary level used is *CF*>10 (12,749 words) and all the documents belonging to 'Easy Ham' (the first category in alphabetical order) are used to calculate the mean and the variance of each word.

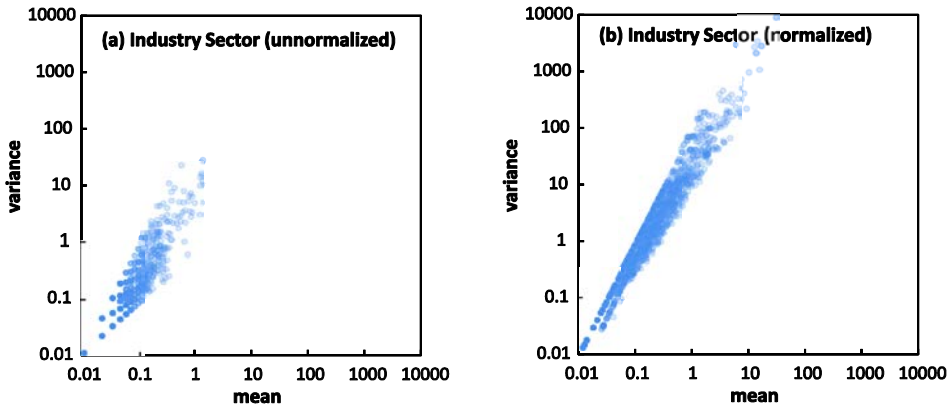


Figure 9: The same as in Fig. 7, but for the Industry Sector dataset. Vocabulary level used is $CF > 10$ (14,317 words) and all the documents belonging to 'accident.and.health.insurance.industry' (the first category in alphabetical order) are used to calculate the mean and variance of each word.

6 CONCLUSION

The usual Poisson distribution and two well-known Poisson mixtures (the K-mixture and negative binomial distributions) have been utilized to build three types of generative probabilistic text classifiers. The classifier frameworks and assumptions used in constructing the classifiers were demonstrated with practical techniques for parameter estimation and document length normalization. The performance of the proposed classifiers was examined through experiments on automatic text categorization of the 20 Newsgroups, SpamAssassin, and Industry Sector datasets. For comparison, a classifier using the multinomial distribution (i.e., the standard multinomial naive Bayes classifier) was also applied to the same datasets.

The results showed that, in the case of non-normalized datasets in which each document length is different from the others, the multinomial naive Bayes classifier performs best but that the classifiers with the K-mixture and negative binomial distributions perform similarly to the multinomial naive Bayes classifier; the Poisson classifier performs worst. On the other hand, the results for normalized datasets, in which each document is normalized to exactly the same length, showed that the three classifiers with the usual Poisson, K-mixture, and negative binomial distributions perform much better than the multinomial naive Bayes classifier. It was also shown from the results for the normalized datasets that the classifier with the Poisson distribution performs best among all the examined classifiers, even though the Poisson model gives a cruder description of term occurrence in real texts than the K-mixture and negative binomial models do.

The origin of the superiority of the Poisson classifier for normalized datasets was discussed in terms of a trade-off between fit and model complexity. Through the discussion, we found that the Bayesian information criterion, which is one of the widely used information criteria,

can qualitatively give a reasonable description of model suitability that is consistent with the classification accuracy of the examined classifiers.

At present, understanding of the relation between the information criteria and the actual classification performance is limited, although our results indicate a strong correlation. We consider that further quantitative analysis is needed before reaching a final conclusion, and such an investigation is planned for future research.

We thank Dr. Yusuke Higuchi for useful discussion and illuminating suggestions. This work was supported in part by JSPS Grant-in-Aid (Grant No. 25589003).

REFERENCES

[1]. K. Church, W. A. Gale, Poisson Mixtures, Natural Language Engineering 1 (1995) 163--190.

[2]. K. Church, W. A. Gale, Inverse Document Frequency (IDF): A Measure of Deviations from Poisson, in: Proceedings of the Third Workshop on Very Large Corpora, 1995, pp. 121--130.

[3]. H. Ogura, H. Amano, M. Kondo, Feature selection with a measure of deviations from Poisson in text categorization, Expert Systems with Applications 36 (2009) 6826--6832.

[4]. H. Ogura, H. Amano, M. Kondo, . Distinctive characteristics of a metric using deviations from Poisson for feature selection, Expert Systems with Applications 37 (2010) 2273--2281.

[5]. H. Ogura, H. Amano, M. Kondo, Comparison of metrics for feature selection in imbalanced text classification, Expert Systems with Applications 38 (2011) 4978--4989.

[6]. S. Katz, Distribution of content words and phrases in text and language modelling, Natural Language Engineering 2 (1996), 15--59.

[7]. J. Gao, M. Li, K. Lee, N-gram distribution based language model adaptation, in: ICSLP2000 Proceedings of International Conference on Spoken Language Processing, 2000, pp. 497--500.

[8]. J. Gao, K. Lee, Distribution-based pruning of backoff language models. in: Proceedings of the 38th Annual Meeting on Association for Computational Linguistics, 2000, pp. 579--588.

[9]. Y. Gotoh, S. Renals, Variable Word Rate N-grams, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing. Vol. 3, 2000, pp. 1591--1594.

[10]. M. Jansche, Parametric Models of Linguistic Count Data, in: Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, 2003, pp. 288--295.

[11]. M. Saravanan, S. Raman, B. Ravindran, A Probabilistic Approach to Multi-document Summarization for Gererating a Tiled Summary, in: ICCIMA '05 Proceedings of the Sixth International Conference on Computational Intelligence and Multimedia Applications, 2005, pp. 167--172.

[12]. M. Saravanan, B. Ravindran, S. Raman, Improving Legal Document Summarization Using Graphical Models, in: Proceedings of the 2006 conference on Legal Knowledge and Information Systems: JURIX 2006: The Nineteenth Annual Conference, 2006, pp. 51--60.

[13]. H. Pande, H. S. Dhami, Distributions of different parts of speech in different parts of a text and in different texts, The modern journal of applied linguistics, ISSN 0974-8741 (2010) 152--170.

[14]. S. Kim, H. Seo, H. Rim, Poisson Naive Bayes for Text Classification with Feature Weighting. in: International Workshop on Information Retrieval with Asian Languages, 2003, pp. 33-40.

[15]. S. Kim, K. Han, H. Rim, H. Myaeng, Some effective techniques for naive Bayes text classification, IEEE transactions on knowledge and data engineering 18 (2006) 1457--1466.

[16]. S. Eyheramendy, D. Lewis, D. Madigam, On the Naive Bayes Model for Text Categorization, in: Proceedings of the 9th International Workshop on Artificial Intelligence and Statistics, 2003, pp. 332--339.

[17]. E. M. Airoldi, W. Cohen, S. E. Fienberg, Statistical Models for Frequent Terms in Text, Tech. Report CMU-CALD-04-106, School of Computer Science, Carnegie Mellon Univ. (2004).

[18]. E. M. Airoldi, A. G. Anderson, S. E. Fienberg, K. K. Skinner, Who Wrote Ronald Reagan's Radio Addresses?, Bayesian Analysis 1 (2006) 289--320.

[19]. T. Mitchell, Machine Learning, McGraw Hill, 1997.

[20]. R. E. Madsen, D. Kauchak, C. Elkan, Modeling word burstiness using the Dirichlet distribution, in: Proceedings of the 22nd International Conference on Machine Learning, 2005, pp. 545--552.

[21]. S. Clinchant, E. Gaussier, The BNB distribution for text modeling, in: Advances in Information Retrieval. 30th European Conference on IR Research, 2008, pp. 150--161.

[22]. B. Allison, An improved hierarchical Bayesian Model of Language for document classification, in: Proceedings of the 22nd International conference on computational linguistics, 2008, pp. 25--32.

[23]. H. Ogura, H. Amano, M. Kondo, Gamma-Poisson Distribution Model for Text Categorization, ISRN Artificial Intelligence Vol. 2013, (2013) Article ID 829630, http://dx.doi.org/10.1155/2013/829630.

[24]. K. Lang, NewsWeeder: Learning to Filter Netnews, in: Proceedings of the 12th International Machine Learning Conference, 1995, pp. 331--339, Morgan Kaufmann.

[25]. N. Slonim, G. Bejerano, S. Fine, N. Tishby, Discriminative feature selection via multiclass variable memory Markov model, Journal on Applied Signal Processing, 2 (2003) 93--102.

[26]. J. Kuha, AIC and BIC - Comparisons of Assumptions and Performance, Sociological Methods and Research, 33 (2004) 188--229.

[27]. S. Konishi, G. Kitagawa, Information Criteria and Statistical Modeling, Springer, 2007.

[28]. M. P. Burnha, D. R. Anderson, Multimodel Inference, Sociological Methods and Research 33 (2004) 261--304.

[29]. M. Ye, P. D. Meyer, S. P. Neuman, On model selection criteria in multimodel analysis, Water Resources Research, 44 (2008) doi:10.1029/2008WR006803.

[30]. P. Tan, M. Steinbach, V. Kumar, Introduction to Data Mining, Addison-Wesley, 2006.

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 2, ISSUE 4
ISSN: 2054 - 7390

Augmenting Weighted Average with Confusion Matrix to Enhance Classification Accuracy

V.Mohan Patro[1], Manas Ranjan Patra[2]
Department of Computer Science, Berhampur University, Berhampur-760007, Odisha, India
[1]vmpatro@gmail.com,[2]mrpatra12@gmail.com

ABSTRACT

Accuracy of a classifier or predictor is normally estimated with the help of confusion matrix, which is a useful tool for analyzing how well the classifier can recognize tuples of different classes. Calculation of classification accuracy of a predictor using confusion matrix for two classed attribute is simple. In case of multi classed attribute, we have to take accuracy of all the classes into consideration, to aggregate them to come with the actual accuracy of the particular classifier or predictor for that particular attribute. Here formulating this, weighted average classification accuracy has been introduced for the overall recognition rate of the classifier, which reflects how well the classifier recognizes tuples of various classes. Classification accuracy is being calculated for the classifiers BayesNet(BN), NaiveBayes(NB), J48 and Decision Table(DT) through weighted average accuracy formulation and the trend of the accuracy values for different number of instances is displayed in tables, which shows the flawless calculation.

Key words: Confusion Matrix, Classifiers, Classification Accuracy, Weighted Average Accuracy.

1 INTRODUCTION

Accuracy of a classifier on a given data set is the percentage of test set tuples that are correctly classified by the classifier. It reflects how well the classifier recognizes tuples of various classes. The error rate or misclassification rate of a classifier *M* can be expressed as 1-*Acc*(*M*), where *Acc*(*M*) is the accuracy of *M* [1].

Most common form of expressing classification accuracy is the error matrix (confusion matrix or contingency table). Error matrices compare, on a class-by-class basis, the relationship between known reference data and the corresponding results of the classification procedure.

The Overall Accuracy is computed by dividing the total number of correctly classified elements (i.e., the sum of the elements along the major diagonal) by the total number of elements in confusion matrix.

DOI: 10.14738/tmlai.24.328
Publication Date: 28th August 2014
URL: http://dx.doi.org/10.14738/tmlai.24.328

Individual Class Accuracy is calculated by dividing the number of correctly classified elements for each class by either the total number of elements in the corresponding column or row.

The Producers Accuracy is the result from dividing the number of correctly classified elements for each class (on the major diagonal) by the number of elements "known" to be of that category.

The User's Accuracy is computed by dividing the number of correctly classified elements in each class (on the major diagonal) by the total number of elements that were classified in that class.

The different types of accuracies like producer's accuracy, user's accuracy, overall accuracy etc. are being calculated with the help of different data and they are being compared [2,3,4,5,6,7,8,9]. In [2], Mittal et al. devised to compare producer and user accuracies on land cover images with the help of expectation-maximization algorithm applying on data provided by JAXA, Japan. A combination of the light detection and ranging (LiDAR) height and intensity data proved to be effective for urban land cover classification [3]. In [4], Samiappan et al. present a Non-Uniform Random Feature Selection (NU-RFS) within a Multi-Classifier System (MCS) framework and experimental results demonstrate the superiority of the proposed approach compared to SVM and RFS. In [5], Experimental results show that a multi-band and multi-level wavelet packet approach can be used to drastically increase the classification accuracy. In [6], a new method is proposed using a data structure called Peano Count Tree (P-tree) for decision tree classification and the accuracy is possessed using the parameters overall accuracy, User's accuracy and Producer's accuracy for image classification methods of object oriented classification, Knowledge Base Classification, Post classification and P-tree Classifier. In [7], a bootstrap method to quantify overall decision tree classification accuracy and confidence is described and the application of this for land use sampling strategies is discussed. Classification of waveforms is being discussed in [8]. In [9], an experiment was conducted to evaluate the differences between rule-based classifications of land cover.

2 CONFUSION MATRIX AND METHODOLOGY

A confusion matrix (*also* known as a contingency table or an error matrix) is a table layout that allows visualization of the performance of a supervised learning algorithm [10]. Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class. All correct guesses are located along the diagonal of the table such that errors can be easily visualized by any non-zero values outside the diagonal.

For a classifier to have good accuracy, ideally most of the tuples would be represented along the diagonal of the confusion matrix (CM). Given two classes, we can introduce the notion of positive tuples (tuples of the class, e.g., *buys computer = yes*) and negative tuples

(e.g., *buys computer = no*). True positives refer to the positive tuples that were correctly labeled by the classifier, while true negatives are the negative tuples that were correctly labeled by the classifier. False positives are the negative tuples that were incorrectly labeled (e.g., tuples of class *buys computer = no* for which the classifier predicted *buys computer = yes*). Similarly, false negatives are the positive tuples that were incorrectly labeled (e.g., tuples of class *buys computer = yes* for which the classifier predicted *buys computer = no*).

2.1 Confusion Matrix for two classes

Table – 1: Confusion matrix

		Predicted Class	
		C_1	C_2
Actual Class	C_1	True positive	False negative
	C_2	False positive	True negative

C_1 – particular class C_2 – different class

True positive (TP) - The number of instances correctly classified as C_1
True negative (TN) - The number of instances correctly classified as C_2
False positive (FP) - The number of instances incorrectly classified as C_1 (actually C_2)
False negative (FN) - The number of instances incorrectly classified as C_2 (actually C_1)

P = Actual positive = TP + FN
P^1 = Predicted positive =TP + FP
N = Actual negative = FP + TN
N^1 = Predicted negative =FN + TN

TP rate = Sensitivity = TP / P = Recall
TN rate = Specificity = TN / N
FP rate = selectivity =1 – TN rate = FP / N
Precision = TP / P^1

Accuracy = (TP + TN) / (P + N)
= TP / (P + N) + TN / (P + N)
= TP / P * P / (P + N) + TN / N * N / (P + N)
= Sensitivity * P / (P + N) + Specificity * N / (P + N)

If a classification system has been trained to distinguish between cats, dogs and rabbits, a confusion matrix will summarize the results of testing the algorithm for further inspection. Assuming a sample of 27 animals — 8 cats, 6 dogs, and 13 rabbits, the resulting confusion matrix could look like the table 2.

Table - 2

		Predicted class		
		Cat	Dog	Rabbit
Actual class	Cat	5	3	0
	Dog	2	3	1
	Rabbit	0	2	11

In this confusion matrix, of the 8 actual cats, the system predicted that three were dogs, and of the six dogs, it predicted that one was rabbit and two were cats. Assuming the confusion matrix above, its corresponding table of confusion, for the cat class, would be:

Table - 3

5 true positives (actual cats that were correctly classified as cats)	3 false negatives (cats that were incorrectly marked as dogs)
2 false positives (dogs that were incorrectly labeled as cats)	17 true negatives (all the remaining animals, correctly classified as non-cats)

From above table classification accuracy for individual class cat can be obtained with the help of the formula for accuracy i.e. (TP + TN) / (P + N). The individual classification accuracy value of cat class will be (5+17) / (5+3+2+17). In this way, 2×2 matrices for dog and rabbit classes can be obtained, from which individual accuracies can be calculated.

The confusion matrix online calculator [11] gives Producer Accuracy, User Accuracy and overall accuracy. The overall accuracy is calculated as the ratio of total of diagonal elements and total elements in confusion matrix. Li Wenkai et al. discussed different formulae like evaluating classification accuracy with positive and background data in their paper [12]. The overall classifier's accuracy has been plotted for different classifiers in paper of Chitra P.K.A. et al.[13].

The overall accuracy of a classifier, in case of multi-classed attribute also, is being calculated as the ratio of total of diagonal elements and total elements in confusion matrix. It means we are taking all the true positive values of all the classes into consideration. In case of two-class attribute, true positive of one class is true negative of another class and vice-versa.

The classification accuracy is (TP + TN) / (P + N)

In our formulation for a multi-classed attribute, all the true negative values of all the classes are being taken into consideration. This means for each of the classes we put the formula of

accuracy to get the individual classification accuracy of the class. Actual count of the particular class is taken as weight for the same class. Aggregating all the individual classification accuracies and weights of all the classes, the weighted average classification accuracy for the attribute is being calculated.

3 CLASSIFICATION TECHNIQUES USED

Bayesian networks are probability based and are used for the reasoning and the decision making in uncertainty, and heavily rely on Bayes' rule. Bayes' rule can be defined as follows [15];

- Assume A_i attributes where i = 1,2,3,…,n, and which take values a_i where i= 1,2,3,…,n.
- Assume C as class label and E = (a_1, a_2,..., a_n) as unclassified test instance. E will be classified into class C with the maximum posterior probability. Bayes' rule for this classification is;

  $P(C \mid E) = \arg\max_c P(C)P(E \mid C)$

Naïve Bayesian Classifier is one of the Bayesian Classifier techniques which is also known as the state-of-the-art of the Bayesian Classifiers. In many works it has been proven that Naïve Bayesian classifiers are one of the most computationally efficient, effective and simple algorithms for Machine Learning and Data Mining applications [16]- [19]. Naïve Bayesian classifiers assume that all attributes within the same class are independent given the class label. Based on this assumption, the Bayesian rule has been modified as follows to define the Naïve Bayesian rule;

$P(C|E) = \arg\max_c P(C) \prod_{i=1}^{n} P(A_i \mid C)$

J48 is an open source Java implementation of the C4.5 algorithm in the WEKA data mining tool. C4.5 builds decision trees from a set of training data in the same way as ID3, using the concept of information entropy [20]. The training data is a set $S = s_1, s_2, \ldots$ of already classified samples. Each sample s_i consists of a p-dimensional vector $(x_{1,i}, x_{2,i}, \ldots, x_{p,i})$ where the x_j represent attributes or features of the sample, as well as the class in which s_i falls.

Decision table is based on logical relationships just as the truth table. It is a tool that helps us to look at the combination of both completeness and inconsistency of conditions [21]. Decision tables, like decision trees or neural nets, are classification models used for prediction. They are induced by machine learning algorithms. A decision table consists of a hierarchical table in which each entry in a higher level table gets broken down by the values of a pair of additional attributes to form another table. The structure is similar to dimensional stacking.

4 WEIGHTED AVERAGE ACCURACY(WAA) ALGORITHM

Weighted average accuracy is being defined as

WAA = $\frac{1}{N} \sum_{i=1}^{n} A_i * C_i$ where

n is no. of possible classes of the multi-class attribute

A_i is individual accuracy for i^{th} class

C_i is instances count for i^{th} class

N is total instances count = $\sum_{i=1}^{n} C_i$

Weighted Average Accuracy Algorithm:
Input: Confusion Matrix
Output: Weighted Average Accuracy
WAA (A, CM)
//CM is n X n confusion matrix, where n is number of classes of an attribute, on which basis //classification accuracy is calculated. A is n+1 X n+3 matrix, where first n X n is filled up with CM
BEGIN
For i=1 to n, A(n+1,i) = $\sum_{j=1}^{n} A(j,i)$ // sum of n columns
For i=1 to n+1, A(i,n+1) = $\sum_{j=1}^{n} A(i,j)$ // sum of n+1 rows, where A(n+1,n+1) is number of instances
For i=1 to n, A(i,n+2) = A(i,n+1) +A(n+1,i) – 2 x A(i,i)
// A(i,n+2) is error i.e. sum of FP & FN and A(i,i) is TP of individual class
For i=1 to n, A(i,n+3) = A(i,n+1) x [1- A(i,n+2) / A(n+1,n+1)]
// A(i,n+3) is individual weighted accuracy, where A(i,n+1) is weight
A(n+1,n+3) = $\sum_{i=1}^{n} A(i,n+3)$ / A(n+1,n+1)
// $\sum_{i=1}^{n} A(i,n+3)$ is total weighted accuracy & A(n+1,n+3) is weighted average accuracy
Return A(n+1,n+3)
END

5 EXPERIMENTATION

The techniques developed in the preceding sections have been applied on two data sets, viz., demographic data and student performance data in undergraduate examinations.

5.1 Data Set 1

This data set consists of demographic profile of citizens (UCI's census dataset) [14]. This dataset has 30162 instances with 15 attributes, such as Age, Work-class, Final-weight, Education, Education-num, Marital-status, Occupation, Relationship, Race, Sex, Native-country, capital-gain, capital-loss, Hours-per-week, and Income. Here, the attribute on which basis the classification accuracy is to be calculated is "education". This attribute is having 16 classes i.e. Bachelors, HS-grade, 11^{th}, Masters, 9^{th}, Some-college, Assoc-acdm, 7^{th}-8^{th}, Doctorate, Assoc-voc, Prof-school, 5^{th}-6^{th}, 10^{th}, Preschool, 12^{th} & 1^{st}-4^{th}. So the classifier will give 16×16 confusion matrix.

5.2 Data Set 2

This data set involves performance of students from different backgrounds (rural/urban/distance learning/regular students) in the university examinations. The data set

has 33254 instances with 10 attributes namely data-year, stream, gender, caste, rururb, gtotal, grtot, tot2, tot1, result. Classification accuracy is computed based on the attribute "result" which has 4 possible values, viz., Fail, Pass, 2^{nd} and 1^{st}. Thus, the classifier will give rise to a 4×4 confusion matrix.

5.3 WEKA Workbench

All simulations were performed in the WEKA (Waikato Environment for Knowledge Analysis) machine learning platform that provide a workbench which consists of collection of implemented popular learning schemes that can be used for practical data mining and machine learning works.

We compare the results of four classifiers BayesNet (BN), NaiveBayes (NB), J48 and Decision Table (DT). The simulations are conducted using two different test options i.e. "Use Training set" and Cross-Validation.

5.4 "Use Training set" and Cross-Validation

The "use training set" option is to train the model with whole training data. In this option, the classifier is evaluated on how well it predicts the class of the instances it was trained on. In cross-validation option, the classifier is evaluated by cross-validation, using the number of folds that are entered in the Folds text field. Here number of folds is 10. Cross-validation calculates the accuracy of the model by separating the data into two different subsets, namely, training set and validation set or testing set. The training set is used to perform the analysis and the validation set is used to validate the analysis. This testing process is continued k times to complete the k-fold cross validation procedure. We have used 10-fold cross-validation. In 10-Fold cross-validation given dataset is partitioned into 10 subsets. From these 10 subsets 9 subsets are used to perform a training fold and a single subset is used as the testing data. The process is repeated 10 times such that each subset is used as a test subset once. The estimated accuracy is then the mean of the estimates for each of the classifiers.

6 IMPLEMENTATION OF WAA ALGORITHM

For data set 1, the confusion matrix will be of 16×16 as "education" attribute is having 16 class values. The Table-4(A) displays the 16×16 confusion matrix data obtained from the classifier for the User profile dataset, having 1000 instances. This is obtained with the help of the classifier Naïve Bayes for the attribute education. It gives 16×16 matrix, because the education attribute is having 16 class values. This same data are in first 16 rows and first 16 columns of the table 5. The accuracy for the bachelors class of attribute education is calculated from the 2×2 confusion matrices i.e. given in table – 4(A & B) (in case of table-4(A), the dark lines give 2×2 matrix). This process is same as obtaining table-3 i.e. confusion matrix for the cat class from table-2 i.e. whole confusion matrix of animals.

Table-4(A)

171	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
0	325	0	0	0	4	0	0	0	0	0	0	0	0	0	0
0	0	37	0	0	0	0	0	0	0	0	0	2	0	0	0
0	0	0	55	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	14	0	0	1	0	0	0	0	0	0	0	0
0	2	0	0	0	216	0	0	0	3	0	0	0	0	0	0
2	0	0	0	0	0	32	0	0	1	0	0	0	0	0	0
0	0	0	0	1	0	0	12	0	0	0	1	0	0	0	0
0	0	0	0	0	0	0	0	15	0	1	0	0	0	0	0
0	0	0	0	0	2	0	0	0	46	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	11	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0	9	0	0	0	0
0	0	1	0	0	0	0	0	0	0	0	0	17	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0
0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	2

Table-4(B)

171 true positives (actual bachelors that were correctly classified as bachelors)	2 false negatives (bachelors that were incorrectly marked as Masters)
2 false positives (Assoc-acdm that were incorrectly labeled as bachelors)	825 true negatives (all the remaining education classes, correctly classified as non- bachelors)

The process of obtaining Table-4(B), 2 × 2 confusion matrix for 'bachelors' class from Table-4(A) is explained as follows. For the 1st class (bachelors), the diagonal element of row-1 & column-1 is the true positive; sum of the other elements of row-1 is false negative; sum of the other elements of column-1 is false positive; sum of rest elements is true negative. To get the confusion matrix for 2nd class (HS-grad), row-2 & column-2 are taken into consideration. This process is applicable for other classes also.

In Table-5 first 16 × 16 matrix data is actual data from the 16 × 16 confusion matrix. 17th row is meant for sum of column elements. 17th column is for sum of row elements. 18th column is for error elements i.e. sum of false positives & false negatives. 19th column is for weighted accuracy. Bold marked value is WAA.

Table-5

171	0	0	2	0	0	0	0	0	0	0	0	0	0	0	173	4	172.31
0	325	0	0	0	4	0	0	0	0	0	0	0	0	0	329	6	327.03
0	0	37	0	0	0	0	0	0	0	0	2	0	0	0	39	3	38.88
0	0	0	55	0	0	0	0	0	0	0	0	0	0	0	55	2	54.89
0	0	0	0	14	0	0	1	0	0	0	0	0	0	0	15	2	14.97
0	2	0	0	0	216	0	0	0	3	0	0	0	0	0	221	11	218.57
2	0	0	0	0	0	32	0	0	1	0	0	0	0	0	35	3	34.90
0	0	0	0	1	0	0	12	0	0	0	1	0	0	0	14	4	13.94
0	0	0	0	0	0	0	0	15	0	1	0	0	0	0	16	2	15.97
0	0	0	0	0	2	0	0	0	46	0	0	0	0	0	48	6	47.71
0	0	0	0	0	0	0	0	1	0	11	0	0	0	0	12	2	11.98
0	0	0	0	0	0	0	1	0	0	0	9	0	0	0	10	4	9.96
0	0	1	0	0	0	0	0	0	0	0	0	17	0	0	18	3	17.95
0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	2	2.00
0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	9	0	9.00
0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	2	4	4	3.98
173	327	38	57	15	222	32	14	16	50	12	12	19	0	9	4	1000		**0.994027**

Formulation of WAA from Confusion Matrix for multi-class attribute :-

Confusion Matrix – n x n → A(n,n)

1	2 …	n	n+1	n+2	n+3
A(1,1)	A(1,2)…..	A(1,n)	$\sum_{i=1}^{n} A(1,i)$		
A(2,1)	A(2,2)……	A(2,n)	$\sum_{i=1}^{n} A(2,i)$		
.					
.					
.					
A(n,1)	A(n,2)…..	A(n,n)	$\sum_{i=1}^{n} A(n,i)$		
$\sum_{i=1}^{n} A(i,1)$	$\sum_{i=1}^{n} A(i,2)$….	$\sum_{i=1}^{n} A(i,n)$			

1. For i=1 to n, A(i,n+1) = $\sum_{j=1}^{n} A(i,j)$ → for n+1 column
2. For i=1 to n, A(n+1,i) = $\sum_{j=1}^{n} A(j,i)$ → for n+1 row
3. For i=1 to n, A(i,n+2) = A(i,n+1) +A(n+1,i) – 2 x A(i,i) → for n+2 column
4. For i=1 to n, A(i,n+3) = A(i,n+1) x [1- A(i,n+2) / A(n+1,n+1)] → for n+3 column, where A(n+1,n+1) is total number of instances.
5. Weighted Accuracy = $\sum_{i=1}^{n} A(i,n+3)$
6. **Weighted Average Accuracy,** A(n+1,n+3) = $\sum_{i=1}^{n} A(i,n+3)$ / A(n+1,n+1)

6.1 Data Set 1

The overall accuracy (OA) i.e. the sum of diagonal elements / the sum of all elements and WAA are being calculated for different classifiers with various numbers of instances. From table-6(A) it is clear that WAA is giving high precision value in comparison to OA.

Table-6(A)

Classifiers	1000 (OA)	1000 (WAA)	30162(OA)	30162 (WAA)
BN	0.959	0.993829	0.99817651	0.99986132
NB	0.971	0.994027	0.99472847	0.99927758
J48	0.998	0.999988	1.00000000	1
DT	0.998	0.999650	1.00000000	1

In Table-6(B) the values for weighted average accuracies as well as overall accuracies are given.

These values are result of the simulations of the classifier Bayes Net for instances 1000, 5000, 10000, 15000, 20000 and 30162. It is clear that WAA values are having high precision and consistency.

Table-6(B)-1

	1000	5000	10000
OA	0.959	0.992	0.9955
WAA	0.993829	0.9961204	0.99967512

Table-6(B)-2

	15000	20000	30162
OA	0.997	0.99765	0.99817651
WAA	0.99973017	0.99980905	0.99986132

The weighted average accuracy is calculated for different number of instances i.e. 1000, 5000, 10000, 15000, 20000, 25000 & 30162 for four classifiers i.e. Bayes Net (BN), Naïve Bayes (NB), J48 and Decision Table (DT) for the test option cross-validation. These values are given in the table 7.

Table-7(A)

	1000	5000	10000	15000
BN	0.993829	0.9961204	0.99967512	0.99973017
NB	0.994027	0.9986268	0.9990493	0.99920068
J48	0.999988	1	1	1
DT	0.999650	1	1	1

Table-7(B)

	20000	25000	30162
BN	0.99980905	0.99983766	0.99986132
NB	0.99920948	0.99926319	0.99927758
J48	1	1	1
DT	1	1	1

In fig-1 the graphs have been plotted for the above values, which show the increasing order of the accuracy is so clear.

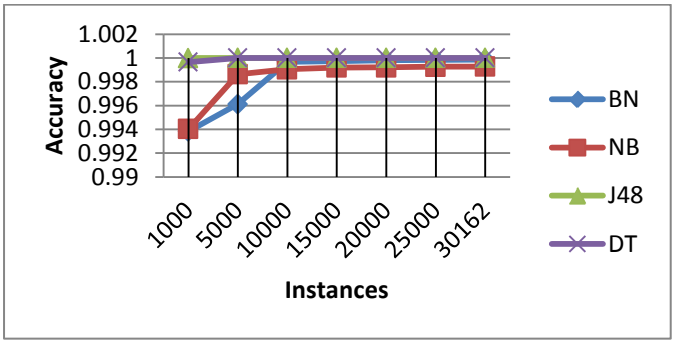


Fig-1

The weighted average accuracy is calculated for different number of instances i.e. 5000, 10000, 15000 & 20000 for the classifier Bayes Net (BN) for the test options use training set (UTS) & cross-validation (CV). The values are given in table 8.

Table-8

BN	5000	10000	15000	20000
UTS	0.99916259	0.99975627	0.99983151	0.99986265
CV	0.9961204	0.99967512	0.99973017	0.99980905

In fig-2 the graphs have been plotted for the above values. In different options also it shows the increasing order of the accuracy is clear.

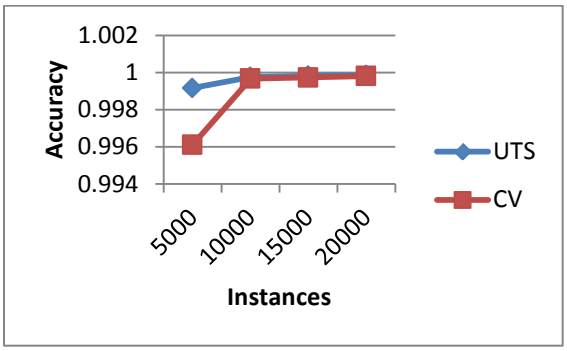


Fig-2

The weighted average accuracy is calculated for different number of instances i.e. 5000, 10000, 15000 & 20000 for the classifiers Naïve Bayes (NB) for the test options use training set & cross-validation . The values are in Table-9 and graphs are in fig-3.

Table-9

NB	5000	10000	15000	20000
UTS	0.99910824	0.99927524	0.999394947	0.999400623
CV	0.9986268	0.9990493	0.999200676	0.999209475

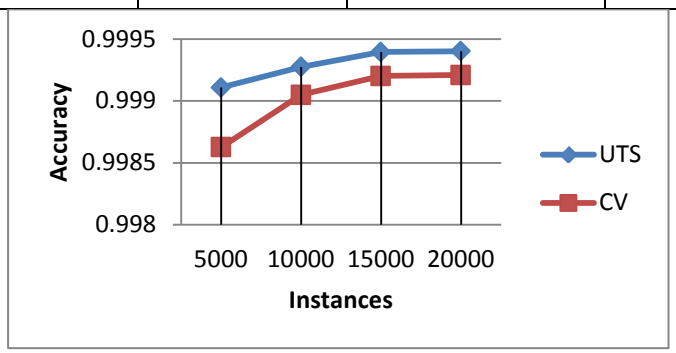


Fig-3

The above process is repeated for the classifier J48 and accuracy values are in table-10 and graphs are in fig-4.

Table - 10

J48	5000	10000	15000	20000
Use training set	1	1	1	1
Cross-validation	1	1	1	1

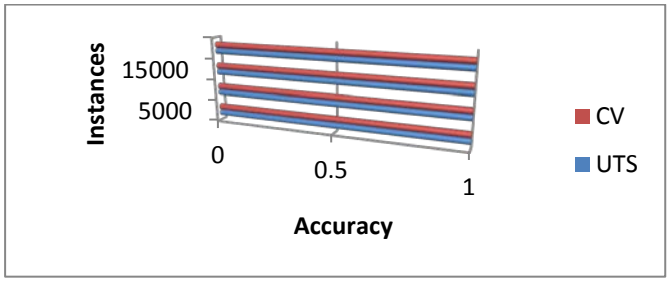


Fig-4

6.2 Data Set 2

In order to verify the accuracy of our proposed algorithm, we have also applied the technique on another data set. Table-11(A), 11(B)-1, and 11(B)-2 summarize the results obtained on the second data set. It is observed that the classifiers behave in a similar fashion confirming that the accuracy increases to a considerable extent as the number of instances grows. Thus, one can establish the supremacy of "Weighted Average Accuracy" technique over the "Overall Accuracy" computation irrespective of the data set used.

Table-11(A)

Classifiers	1000 (OA)	1000 (WAA)	33254 (OA)	33254(WAA)
BN	0.787	0.890326	0.868106	0.92404887
NB	0.794	0.894396	0.855266	0.915228077
J48	0.783	0.88757	0.880014	0.932542206
DT	0.779	0.885846	0.866633	0.923259574

Table-11(B)-1

	5000	10000	15000
OA	0.7938	0.8276	0.844333
WAA	0.89446528	0.90468752	0.912737067

Table-11(B)-2

	20000	25000	30000
OA	0.85485	0.8602	0.863867
WAA	0.917283103	0.919080608	0.921331164

7 CONCLUSION

The aim of our work was to enhance the accuracy of any classifier. Towards this end, we have formulated a technique called weighted average accuracy which is obtained by aggregating the individual accuracies for all class values of the particular attribute using the weight factor. The WAA takes the number of particular class in an attribute as the weight factor to calculate the classification accuracy. Individual accuracy is calculated with this weight factor. Lastly average of the total weighted accuracy is taken as final value. From both the data sets it is observed that for any number of instances, for any classifier, WAA out performs OA.

REFERENCES

[1] Jiawei Han and Micheline Kamber, *Book on"* Data Mining: Concepts and Techniques*", 2nd ed.,* Morgan Kaufmann Publishers, March 2006. ISBN 978-1-55860-901-3.

[2] Vikas Mittal, D. Singh and L.M. Saini; "Land Cover Classification using EM Algorithm based Multi-Polarized ALOS PALSAR Image Fusion"; IEEE, 2013, Page(s) 5pp.

[3] Weiqi Zhou; "An Object-Based Approach for Urban Land Cover Classification Integrating LiDAR Height and Intensity Data"; IEEE, 2013, pp.928-931.

[4] Sathish kumar Samiappan, Saurabh Prasad and Lori M Bruce; "Non-Uniform Random Feature Selection and Kernel Density Scoring With SVM Based Ensemble Classification for Hyperspectral Image Analysis"; IEEE, 2013, pp. 792-800.

[5] S. Rajesh and S. Arivazhagan ; "Land CoverLand Use Mapping using Different Wavelet Packet Transforms for LISS IV Imagery"; IEEE, 2011, pp. 103-108.

[6] M. Seetha, K. V. N. Sunitha, D.V. Lalitha Parameswari, G. Ravi; "Accuracy Assessment of Object Oriented and Knowledge Base Image Classification using P-Trees"; IEEE, 2010, pp. 760-763.

[7] Catherine Champagne, Heather McNairn, Bahram Daneshfar, Jiali Shang; "A bootstrap method for assessing classification accuracy and confidence for agricultural land use mapping in Canada"; Elsevier, 2014, pp. 44-52.

[8] Karolina D. Fieber, Ian J. Davenport, James M. Ferryman, Robert J. Gurney, Jeffrey P. Walker, Jorg M. Hacker, "Analysis of full-waveform LiDAR data for classification of an orange orchard scene"; Elsevier, 2013, pp. 63-82.

[9] Mariana Belgiu, Lucian Dra˘gut, Josef Strobl; "Quantitative evaluation of variations in rule-based classifications of land cover in urban neighborhoods using WorldView-2 imagery"; Elsevier, 2013, pp. 205-215.

[10] http://en.wikipedia.org/wiki/Confusion_matrix last accessed on 10/04/13.

[11] http://www.dicom.uninsubria.it/~marco.vanetti/cfmatrix last accessed on 15/05/13.

[12] Wenkai Li and Qinghua Guo, "A New Accuracy Assessment Method for One-Class Remote Sensing Classification", IEEE, 2013, pp. 1-12.

[13] P.K.A. Chitra and S. Appavu Alias Balamurugan, "Benchmark Evaluation of classification methods for single label learning with *R",* IEEE, 2013, pp. 746-752.

[14] Asuncion A. and Newman D.J. (2007) UCI Machine Learning Repository Irvine, CA: University of California, School of Information and Computer Science. [Online] Available from: http://www.ics.uci.edu/~mlearn/MLRepository.html last accessed on 11/02/13.

[15] Jensen F.V. "Introduction to Bayesian Networks*"*. Denmark: Hugin Expert A/S, 1993.

[16] Wang Z. and I. Webb G., "Comparison of lazy bayesian rule and tree-augmented bayesian learning", IEEE, 2002, pp. 490 – 497.

[17] Shi Z., Huang Y. and Zhang S., "Fisher score based naive Bayesian classifier", IEEE, 2005, pp. 1616-1621.

[18] Xie Z. and Zhang Q., "A study of selective neighborhood-based naïve bayes for efficient lazy learning". 16th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2004.

[19] Santafe G., Loranzo J.A. and Larranaga P., "Bayesian model averaging of naive bayes for clustering", IEEE, 2006, Page(s) 1149 – 1161.

[20] http://en.wikipedia.org/wiki/C4.5_algorithm last accessed on 21/03/13.

[21] Alicia Y.C. Tang, Nur Hanani Azami and Norfaezah Osman, "Application of Data Mining Techniques in Customer relationship Management for An Automobile Company", IEEE, 2011, Page(s) 7pp.

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 2, ISSUE 4
ISSN: 2054 - 7390

Fuzzy Logic Approach for Person Authentication Based on Palm-print

Rajkumar Mehar[1], Kapil Kumar Nagwanshi[2]

[1]*Computer Science & Engineering Department, Rungta College of Engineering & Technology, Kohka Kurud Road, Bhilai*

[1]rajmeharmehar24@gmail.com, [2]kapilkn@ieee.org

ABSTRACT

The aim of present research is to classify the palmprint images using fuzzy logic for different security aspects. Fuzzy logic is relatively efficient and advanced theory, at present wide use of fuzzy logic is in the classification of remotely sensed images. A branch of biometric, palmprint authentication have increasing attention because palmprint is unique, permanent, measurable characteristics having voluminous of the line features. In this paper we discuss a method for feature extraction, identification (recognition) techniques of palmprints based on fuzzy logic technique and some publically available databases. This method use sub image based principle line feature extraction technique in low resolution palmprint images. Image is divided into sub images and feature obtained from these subimages are combined to generate a single feature vector for the palmprint image. This vector is provides to fuzzy inference system as input. The testing for system has been performed on IITD, and PolyU databases. Experiments were carried to show the effectualness of our proposed approach have an accuracy of 89.46%.

Keywords*:* Biometric, ROI, Fuzzy Inference System, Membership functions, if-then rules.

1 INTRODUCTION

For security system, the authentication and identification of a person is carry out with signature, user id, password and cards. These techniques are not enough at present days because signature can be replicated, passwords can be guessed and cards can be misplaced or stolen So, biometric features are used to recognize an individual. The recognition of an individual is performed via two steps identification (validation) and authentication (verification) process. The biometric features are consists of behavioural characteristics (heart-beat, voice) and anatomical (fingerprints, face, palm, iris) [1]. The palmprint is best for the recognition of an individual. Palmprint is a pattern of principle lines ridges wrinkles and minutie points. Each and every individual has its unique and different palmprint. In this paper we discuss about image segmentation, feature extraction, identification and verification (recognition) techniques and some publically available databases.

DOI: 10.14738/tmlai.24.378
Publication Date: 28th August 2014
URL: http://dx.doi.org/10.14738/tmlai.24.378

2 IMAGE PREPROCESSING AND SEGMENTATION

Preprocessing is the procedure to align palmprint position and segment the central area for feature generation. Palmprint segmentation is an important pre-processing step in automatic biometric authentication system. Researchers use four different types of sensors: digital scanners, digital cameras, video cameras and CCD based scanners to gather palmprint images. CCD based palmprint scanner aligns palm accurately and scan biometric traits to collect high quality images. Scanners based on CCD employs pegs to restrict palm orientation for guiding the position of hands [2]. The problem arises in contactless scanning is image quality as image quality is low, cause recognition problem. To establish coordinate system major techniques utilize the key points in between fingers. Preprocessing involve five common processes: (1) palm print images binarization, (2) extracting the contour of palmprint, (3) key point detection, (4) coordination system establishment and (5) ROI extraction, the central part of image [3]. Fig 1(h) shows a preprocessed image. Majority of the preprocessing algorithms select ROI of square area but few of them select ROI of half elliptical and circular area for feature extraction. The square shaped region of extracted ROI is easy to handle translation operation. Half elliptical and circular shaped ROI are easy to handle rotation to the palmprint image.

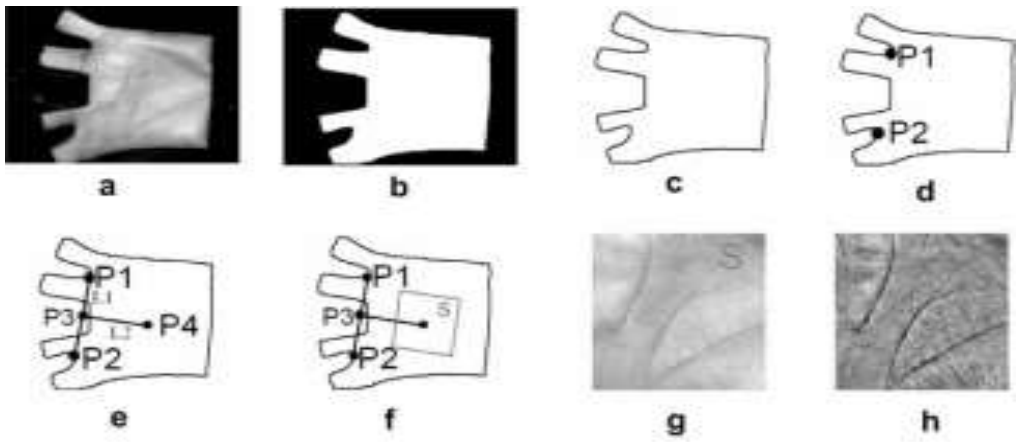


Figure 1: Palmprint Image Peprocessing. (a) Original Image, (b) Image Binarization, (c) Boundry Tracking, (d) Key Point Detection, (e) Establish Coordinate System, (f) Select Central Area, (g)segmented ROI (h) ROI Extracted Image.

3 FUZZY INTERFERENCE SYSTEM

Fuzzy inference is the process, using fuzzy logic to formulate the mapping from a given input to an output [1]. The fuzzy inference process involves three main factors membership functions, if-then rules and fuzzy logic operators. Image processing in fuzzy inference system (FIS) is consecutive task of Image fuzzification, membership functions & rules creation and image defuzzification. There are two types of fuzzy inference systems that can be implemented in the Fuzzy Logic Toolbox:

- Sugeno-type.

- Mamdani-type.

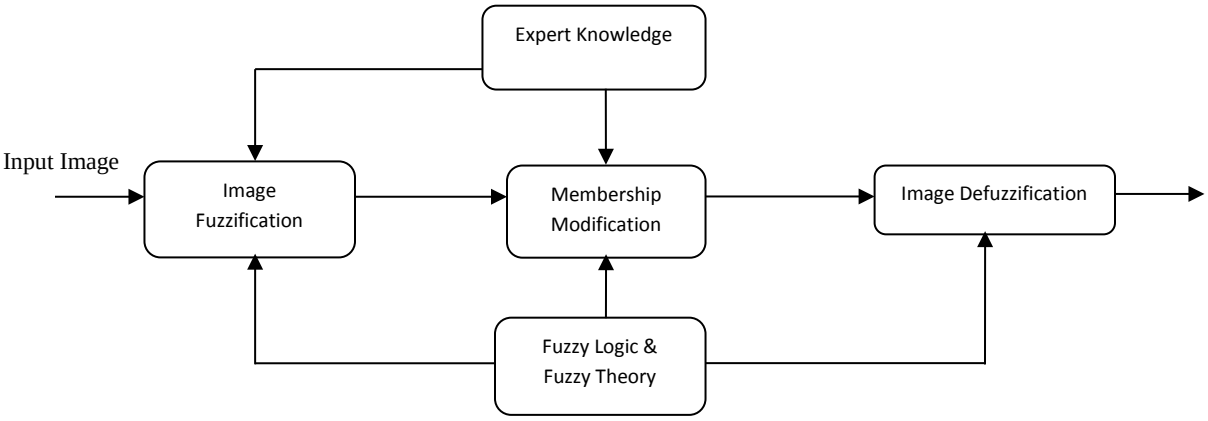


Figure 2: Fuzzy Inference System

Mamdani type fuzzy inference method expects the output membership functions are to be a fuzzy set. It is the most commonly used fuzzy methodology. For each output variable there is a fuzzy set, necessary to aggregate and defuzzified as shown in above figure 2 [4]. A Sugeno-type system is useful to model an inference system having the output membership functions are either linear or constant. Fuzzy inference system of sugeno type is used in recognition in this paper. There are 100 inputs to the fuzzy inference system having range from 0 to 255. One output variable is designed with range from 0 to 500. Image is segmented into 100 blocks, generates one feature value. These feature value of all blocks is considered as input to each input variable. Input variable consist a number of membership function of triangular function (trimf), all membership function is same type. Output variable also have membership function of constant type one for each class. FIS consist 100 rules for recognition purpose by setting up rule weight to 1 and using AND fuzzy operator to connect antecedent of the rule. Figure 3 shows the rule editor of FIS system.

Necessity of FIP: there are many reasons for use of fuzzy techniques in image processing. The most important is in many image-processing applications, it is necessary to use expert knowledge to avoid the difficulties (e.g. scene analysis, object recognition). Fuzzy logic and Fuzzy set theory provides us a powerful tool to represent the human knowledge in the form of fuzzy if–then rules. On the other side due to uncertain data, many difficulties in image processing arise. However, this uncertainty is not always due to randomness of data but to the ambiguity and vagueness of data.

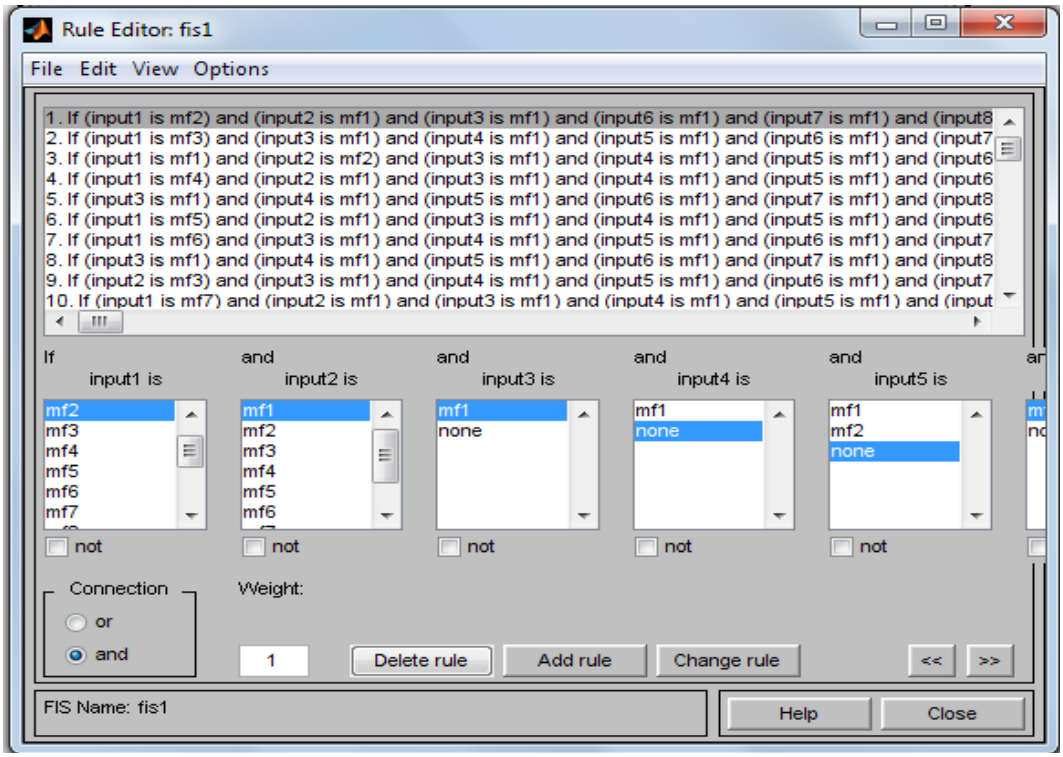


Figure 3: Sample of the Fuzzy Logic Rules Editor

4 FEATURE EXTRACTION

Once the central part is segmented in preprocessing, next step is to extract features for matching operation. Two different types of algorithms are used for recognition, verification and identification. Typical algorithms for Verification are line based, subspace based and statistic based. Line–based approach either uses an existing edge detection method or develops an edge detection method to extract palm line [9,10]. These line are matched either directly or represented in other formats for matching.

Canny edge detector [5] used to detect palm line as shown in figure 4 (b) below. The orientation of the edge point is passed into four membership function representing four directions and finally Euclidian distance is used for pattern matching.

In Subspace-based methods also called appearance based methods in the literature of the recognition. They use principal component analysis (PCA), linear discriminant analysis (LDA), and independent component analysis (ICA) [11]. The subspace coefficient is regarded as features. In subspace method various distance measure and classifier are used to compare the features. In addition to apply PCA, LDA, and ICA directly to palmprint images, researchers also use wavelets [12], Gabor, discrete cosine transform (DCT) and kernel in their method.

Statistical based methods are either local or global statistical approaches. Local statistical approaches transform images into another domain and then divide the transformed images are divided into several small regions. Local statistics for example means and variances of each small region are calculated and regarded as features. To our survey knowledge, no one has yet

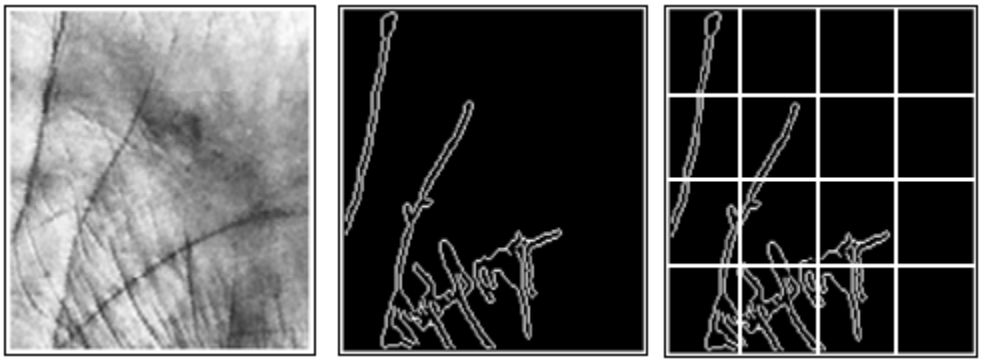
investigated high order statistics for these approaches. The small regions are commonly square but some are elliptical and circular.

In this paper image is filtered for edge detection and then segmented into small regions known as blocks, fig 4 (c) demonstrates the segmentation of palmprint image. After image filtering each point has an intensity value greater than 0 if it is a line element otherwise intensity value is zero.

Feature extraction algorithm is as described below:

- Filter the image with Canny Edge Detection Filter to detect the principle lines with a threshold value.
- Segment the filtered image into sub images (blocks) and Find the no of line element in each block.
- Extract one feature from each block by finding the average intensity value of pixels which is a line element.
- Then features from all sub images are aggregated into a single string to generate feature vector of the image.

Image have different resolution in different dataset so, at first images are resized to 150*150 resolution then segmented to 100 blocks of 15*15 resolution.



(a) (b) (c)

V1	V2	....	V10
V11	V12	....	V20
....	....	....	....
V91	V92	....	V100

[V1, V2, V3, V4, V5,.....V100]

(d) (e)

Figure 4: Feature Extraction. (a) Palmprint Image, (b) Palmline Detection using Canny Filter, (c) Segmentation of Image, (d) Average Intensity Value of Each Blocks, (e) Feature Vectore of image.

5 RECOGNITION

The process of image matching or classification is to compute the degree of similarity between the input test image and a training image from database. In this project the main purpose of fuzzy logic is to perform classification task. For recognition purpose lots of different classifiers have been designed alike SVM, Neural Network (NN), KNN, GMM, LDA, HMM. Fuzzy classification is performed effectively by using and modifying two component, Membership function and Fuzzy logic rules. Fuzzy rules are in the form of “if then else”.

If X1 is A1 and X2 is A2 Then Y is B.

Where X1 and X2 are fuzzy variables and A1 and A2 are fuzzy values. The if part of the rule "X1 is A1" is known as premise or antecedent, and then part of the rule "Y is B" is known as conclusion or consequent. Statements in the antecedent (or consequent) section of the rules can employ fuzzy logical connective such as "OR" and "AND". In the if-then rules, the word "is" have used in entirely two dissimilar ways depends on whether it comes alongs in the antecedent part or in the consequent part.

The image used for testing is either enrolled or not. If image is enrolled then output is a value from 1 to 100 because 100 rules are created. Otherwise image does not belong to any rule and output value is mid range of output variable. For example if image is of second class then rule 2 is executed and output value is 2. If image is not a genuine image then output value is 250 as range of output variable is 0 to 500. Fuzzy rule consists 103 columns 100 for input variables to take input from each block, 1 for output variable to generate the output value, 1 for fuzzy operator which is set to 1 for AND fuzzy operator and last column for fuzzy rule weight range from 0 to 1, here rule weight is set to 1.

6 AVAILABLE DATABASE

Three palmprint image databases are publicly available for the research purpose; these databases are namely CASIA palmprint database, PolyU palmprint database and IIT Delhi palmprint database. The CASIA palmprint image database consist total 5502 palmprint images of 312 subjects [6]. For each person, the left and right palm images have been captured. All the palmprint images have been captured by their self developed “palmprint recognition device”. These palm images are in JPEG format of 8 bit gray level files. This self developed device does not have any pegs to restrain the postures and positions of the palmprints.

PolyU palmprint database consists 500 different palms with 12 samples for one illumination and thus having total 6000 images [7]. The images were accumulated from 55 females and 195 males in two different academic sessions. The images are collected from the person having age between 20 to 60 years. In both sessions, 6 palmprint samples are taken for every palm. PolyU database utilize pegs to avoid the hand orientation. Therefore, this database is used widely as utilization of pegs helps to acquire significantly higher performance.

IIT Delhi palmprint dataset is developed from the image of staff member and students of IIT Delhi during Jul 2006 to Jun 2007 [8]. IITD dataset uses a touch less image acquisition setup. Database consists 235 users palm image, is stored in bitmap (*. bmp) format. For each person 7 images are captured of both left and right hand palm in different hand variation. The touch less imaging results in higher image scale variations. The captured images have 800 X 600 pixels resolution, segmented or cropped image in normalised form of 150 X 150 resolutions are also available. In this paper we use four dataset, IIT DELHI Palmprint database of Left Hand is dataset1, IIT DELHI database of Right Hand is dataset2, PolyU 2D+3D Palmprint Database is dataset3, PolyU Multispectral Palmprint Database of Blue Illusion is dataset4. For testing process 225 images are taken from dataset1 470 images from dataset2, 570 images from dataset3 and from dataset4 of blue illumination 792 images are used.

7 RESULT & DISCUSSION

User interface of proposed system is shown in figure 5. At first palm print recognition system is trained by selecting the training data from any dataset as in figure 6 (a). Thereafter fuzzy inference system is designed on the basis of features extracted from the training dataset. Thereafter testing data is selected to test the system accuracy from corresponding dataset. Figure 6 (b) shows the testing of data to the system. Based on the methodology four dataset are tested in different threshold value and results are obtained as shown in table1. It is cleared from table1 that dataset 2 have the highest accuracy 89.46 and also in different threshold value dataset 2 have best result. Figure7 show the line graph of accuracy to all four dataset at different threshold value. Table 2 shows the comparison of template size in byte of different methods. System speed of various methods in dataset1 is shown in table 3(a) and system speed of various methods in PolyU palmprint dataset is shown in table 3 (b).

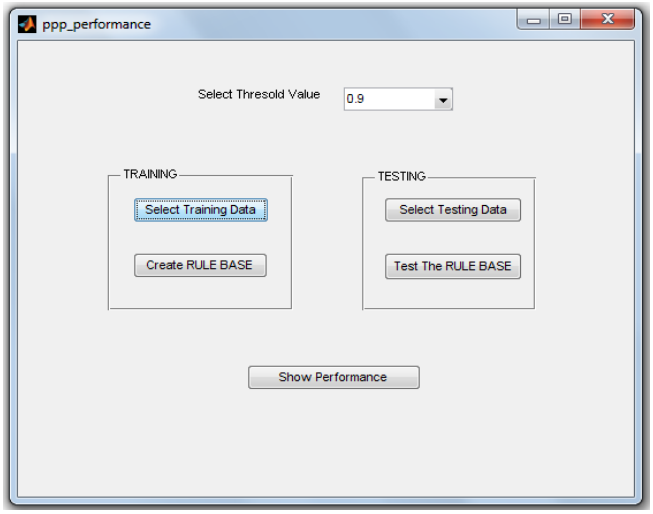
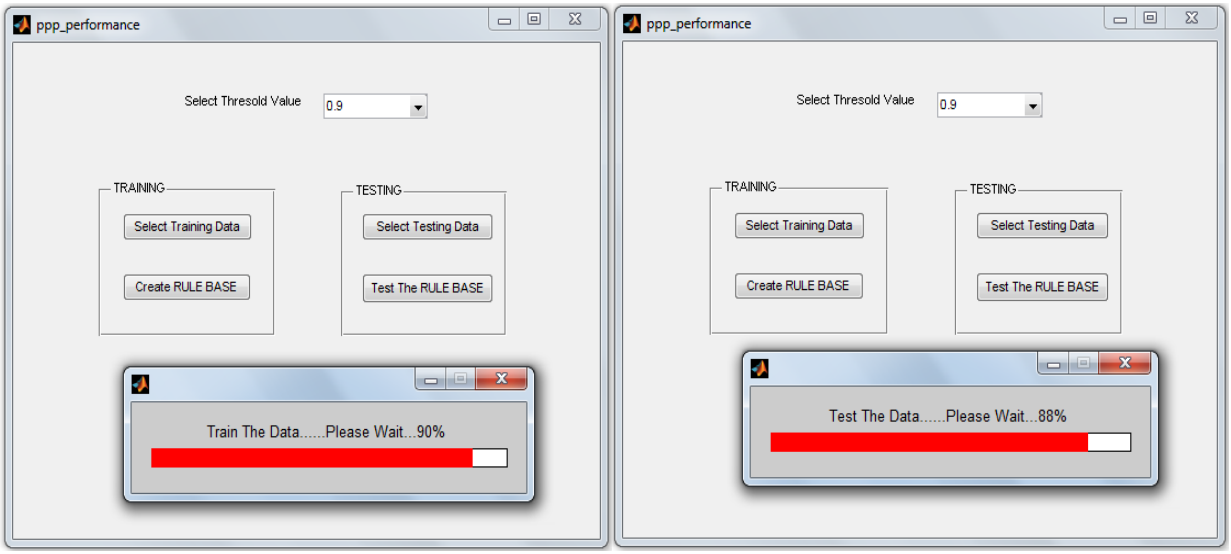


Figure 5: GUI of Palmprint Recognition System



(a) (b)

Figure 6: Identification and Verification process (a) Training to the system, (b) Testing to the system.

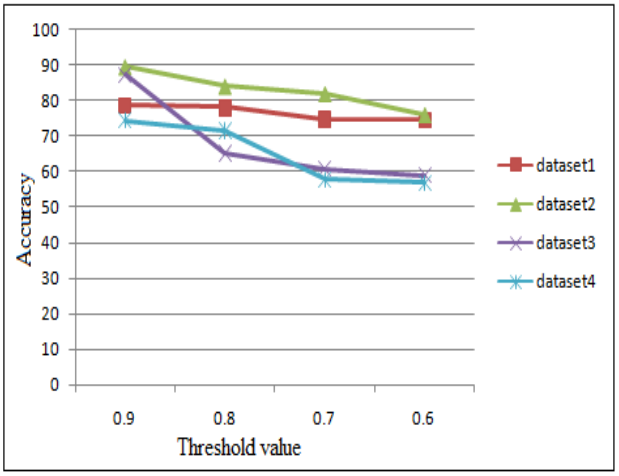


Figure 7: Accuracy of the different datasets

Table 1: Verification accuracies of different databases

Threshold value	Dataset1	Dataset2	Dataset3	Dataset4
0.9	78.6	89.46	87.19	74.30
0.8	78.0	83.93	65.08	61.48
0.7	74.88	81.80	60.78	57.57
0.6	74.67	75.85	58.94	56.88

Table 2: comparison of template size

Method	Template size (byte)
Competitive code	384
Palm code [13]	256
2D SAX [17]	256
This paper	100

Table 3: comparison of speed

(a) System speed for dataset 1

Method	Extraction	Matching	Total
Palmcode [13]	336.1316	2.7186	338.8502
Ordinal code [14]	672.9061	7.9563	680.8624
Kipsang [15]	15.813	1.4971	17.3101
Dale et al [16]	12.343	0.0100	15.3530
This paper	69.9391	67.6674	137.6089

(b) System speed for PolyU palmprint dataset

Method	Extraction	Matching	Total
Palmcode [13]	54.4166	0.73156	55.1482
Ordinal code [14]	155.1370	2.3652	157.5022
Kipsang [15]	14.4190	0.4813	14.9003
Dale et al [16]	5.3010	0.0100	5.3110
This paper	288.9478	201.1187	490.0753

8 CONCLUSION

Salient features of the proposed technique include a low resolution image for feature representation of the palmprint texture, low computational overheads having recognition accuracies over 89%. Accuracies held over 100 classes having 5 samples per class have shown it to be an acceptable and optimized method providing comparable recognition efficiency. Future work would involve testing on more number of classes and fusion of other biometric features like colour of palm, face, or other biometric trait to improve recognition accuracies. This paper also does not include any preprocessing steps i.e. the images are taken from the publicly available dataset as it is.

REFERENCES

[1]. Davide Maltoni, Dario Maio, Anil K. Jain, Salil Prabhakar, *Handbook of Fingerprint Recognition*, Springer Press 2nd Edition 2009.

[2]. C. C. Han, *A* hand-*based personal authentication using a coarse-to-fine strategy*, Image and Vision Computing, 2004. 22 (11): p. 909–918.

[3]. S. S. Hatkar, Sneha M. Ramteke, *Segmentation of Palmprint into Region of Interest*, IJCT ISSN 2277-3061, March-April 2013. 4(2).

[4]. G. Prasanna Lakshmi, J. A. Chandulal, Y. Gopala Krishna, *Finger print Analysis and Matching using fuzzy logic design*, IOSR Journal of Computer Engineering (IOSRJCE) ISSN: 2278-0661, July-Aug 2012. 1(6): p. 04-08.

[5]. J. Canny, "*A* computational *approach to edge detection*, IEEE Transactions on Pattern Analysis and Machine Intelligence*,* 1986. **8**(6): p. 450–463.

[6]. CASIA Palmprint Database, http://biometrics.idealtest.org/dbDetailForUser.do?id=5.

[7]. PolyU Palmprint Database, "http://www.comp.polyu.edu.hk/biometrics".

[8]. IITD Touchless Palmprint Database. Version 1.0 (available online): http://web.iitd.ac.in/~ajaykr/Database_Palm".htm. http://www4.comp.polyu.edu.hk/~csajaykr/IITD/Database_Palm.htm.

[9]. D.S. Huang, W. Jia, D. Zhang, *Palmprint verification based on principal lines*, Pattern Recognition, 2008. 41 (4): p. 1316–1328.

[10]. M.K.H. Leung, A.C.M. Fong, H.S. Cheung, *Palmprint verification for controlling access to shared computing resources*, IEEE Pervasive Computing, 2007. 6 (4): p. 40–47.

[11]. M.K.H. Leung, A.C.M. Fong, H.S. Cheung, *Palmprint verification for controlling access to shared computing resources*, IEEE Pervasive Computing, 2007. 6 (4): p. 40–47.

[12]. M. Ekinci, M. Aykut, *Palmprint recognition by applying wavelet subband representation and kernel PCA*, Lecture Notes in Artificial Intelligence, 2007. p. 628–642.

[13]. D. Zhang, A. W. Kong, J. You, M. Wong, *Online palmprint identification*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 2003. 25(9): p.1041–1050.

[14]. Z. Sun, T. Tan, Y. Wang, *Ordinal palmprint representation for personal identification*, in: Computer Vision and Pattern Recognition, 2005. p. 279–284.

[15]. H. K. Choge, T. Oyama, S. Karungaru, S. Tsuge, M. Fukumi, *Palm- print recognition based on local dct feature extraction*, in: International Conference on Neural Information Processing, 2009. p. 639–648.

[16]. M. P. Dale, M. A. Joshi, N. Gilda, *Texture based palmprint identification using dct features*, International Conference on Advances in Pattern Recognition, 2009. pp. 221–224.

[17]. Jiansheng Chen, Yiu-Sang Moon, Ming-Fai Wong, Guangda Su, *Palmprint authentication using a symbolic representation of images*, Image and Vision Computing, 2010. 28: p.343–351.

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 2, ISSUE 4
ISSN: 2054 - 7390

Parallelization of Termination Checkers for Algebraic Software

Rui Ding, Haruhiko Sato, Masahito Kurihara
Graduate School of Information Science and Technology, Hokkaido University, JAPAN;
{ray,haru}@complex.ist.hokudai.ac.jp, kurihara@ist.hokudai.ac.jp

ABSTRACT

Algebraic software is modeled as a set of equations representing its specification, and when each equation is directed either from left to right or from right to left, the resultant set of directed equations (or rewrite rules) is called a term rewriting system, which can be interpreted as a functional program executed by the pattern matching and term rewriting. In the field of formal verification of information systems, most of the properties of such a system are formalized as inductive theorems, which are equations over terms which hold on recursively-defined data structure such as natural numbers, lists and trees. Well-known as a method for inductive theorem proving is the Rewriting Induction (RI) proposed by Reddy. Recently, this method was extended by Sato and Kurihara to the Multi-context Rewriting Induction with termination checker (MRIt), which is a variant of RI to try to find a suitable context for induction automatically. However, MRIt should perform a large amount of termination checking of term rewriting systems, causing a significant efficiency bottleneck. In this paper, we propose a method of parallelizing the termination checkers used in MRIt to improve its efficiency by focusing on the well-known typical termination checking method based on the lexicographic path orders. For implementation, we used the functional concurrent programming language Erlang. We discuss the efficiency of our implementation based on the experiments with the standard set of termination problems.

Keywords: Algebraic Software, Term Rewriting System, Termination, Parallelization.

1 INTRODUCTION

Algebraic software is modeled as a set of equations representing its specification, and when each equation is directed either from left to right or from right to left, the resultant set of directed equations (or rewrite rules) is called a term rewriting system [1]. A term rewriting system is a set of rewrite rules used for rewriting a term to another, and can be interpreted as a functional program executed by the pattern matching and term rewriting. Given a term rewriting system, we always concern about its termination. A software tool which checks its termination is called a termination checker. The termination property is a very important

DOI: 10.14738/tmlai.24.368
Publication Date: 28th August 2014
URL: http://dx.doi.org/10.14738/tmlai.24.368

property especially in automated inductive theorem proving, which tries to prove inductive theorems, which are equations over terms which hold on recursively-defined data structures, such as natural numbers, lists and trees. Once we have found that the system is terminating in the inductive theorem proving, we can use the transitive closure of the associated rewriting relation as a well-founded order over terms for the basis of induction.

To automate the inductive theorem proving, a lot of methods have been proposed. Here we only mention one of the most refined ones, i.e., the Rewriting Induction (RI) proposed by Reddy [2]. The RI is a principle which has successfully generalized and refined several procedures proposed so far for proving inductive theorems based on term rewriting systems. The RI relies on the termination of term rewriting systems created from the axiomatic equations. However, there are some strategic issues coming from the non-determinism in constructing proofs. The most critical issue is that, in the proof procedure of the RI, we have to choose appropriate proof steps, considering which reduction order should be employed to prove the termination and which rules should be employed for rewriting. In general, it is difficult to choose appropriate strategies leading to the success, because we do not know the final result beforehand. In the RI, the issue of choosing the reduction order is fixed before starting the procedure, by specifying a particular reduction order used to decide the direction of equations to transform them into rewrite rules and ensure that the resultant term rewriting systems have the termination property. However, it is most difficult to properly decide such a particular reduction order beforehand, making the RI really hard to automate.

To solve this problem, Aoto [3] proposed a variant of the RI, named the Rewriting Induction with termination checker (RIt). This method has enabled researchers to improve the efficiency of the inductive theorem proving systems by customizing the external termination checkers, instead of using the reduction order given beforehand in the built-in termination checkers. However, although the RIt has solved some strategic issues mentioned above, there comes another issue instead: in which direction equations should be directed. From the viewpoint of the strategy, the use of the external termination checkers gives us more flexibility in the direction strategy, because the dynamic decision on the direction contributes to the increase in the possibility of the success of the theorem proving. Thus in order to prove inductive theorems automatically, we can now exploit this flexibility by trying various direction strategies in parallel. However, it is clear that if we physically created and ran a number of parallel processes in a naive parallelization scheme, it would cause serious inefficiency.

Recently, Sato and Kurihara [4] proposed a new variant of the rewriting induction procedures called the Multi-context Rewriting Induction with termination checkers (MRIt) based on the idea of multi-completion of Kurihara and Kondo [5] . The procedure simulates the execution of parallel RIt processes in a single process. There are inductive theorems which are easily proved by the MRIt but were not proved by the standard RI or RIt unless the strategies and contexts were chosen correctly or else auxiliary lemma were discovered and supplied. The

MRIt improved the efficiency of inductive theorem proving significantly. However, a large amount of rapid check of termination is necessary in the MRIt. This causes the standard termination checker to take a lot of time for calculation, especially when the checker is based on the dependency pair method, one of the most powerful methods recognized in the associated community, proposed by Arts and Giesl [6]. This is becoming the obstacle for further improvement of its efficiency. In order to automate and accelerate the MRIt, we propose in this paper the use of the multi-core CPU to parallelize the lexicographic path order method, a well-known termination checking method implemented in a lot of termination checkers. We discuss the problem from two viewpoints. One is the exploration of the lexicographic path orders, and the other is a large amount of term rewriting systems to be checked. For the implementation, a functional concurrent programming language named Erlang has been adopted.

The paper is organized as follows. In Section 2, we will briefly review the basic definitions on term rewriting systems. Then we will present our parallelization method in Section 3, and discuss its performance in Section 4. Finally we will come to the conclusion and discuss our future work in Section 5.

2 PRELIMINARIES

Let us briefly review the basic definitions and notations for term rewriting systems (TRSs). In TRSs, a term will be built from function symbols and variables in the usual way. For example, if f is a binary function symbol and x and y are variables, then $f(x, y)$ is a term. To make clear which function symbols are available in a certain context, you need to specify a signature as defined below.

Definition 1: A *signature* Σ is a set of function symbols, where each $f \in \Sigma$ is associated with a non-negative integer *n*, the arity of f. The elements of Σ with arity *n*=0 are called *constant symbols*.

For example, if we want to talk about a group, a well-known algebraic structure equipped with an identity element e, a unary inversion operation i and a binary multiplication operation f, we use the signature $\Sigma = \{e, i, f\}$, where e has arity 0, i is unary, and f is binary. If we want to talk about the set of non-negative integers, we may use the signature consisting of the smallest non-negative integer 0, the successor function *s* (meaning *s*(*x*)=*x*+*1*), and some arithmetic functions such as $+$ and $\times$.

With the definition of signature, we can define terms as follows.

Definition 2: Let Σ be a signature and X be a set of variables such that $\Sigma \cap X = \{\}$. The set $T(\Sigma, X)$ of all Σ*-terms over* X (or simply *terms* if Σ and X are clear from the context) is inductively defined as

- $X \subseteq T(\Sigma, X)$ (i.e., every variable is a term)

- If $t_1, t_2 \ldots, t_n \in T(\Sigma, X)$ and $f \in \Sigma$, then $f(t_1, t_2, \ldots, t_n) \in T(\Sigma, X)$, where n is the arity of f (i.e., application of a function symbol to argument terms yields a term).

For example, for the signature $\Sigma = \{f, g\}$ with two binary function symbols, $f(x, g(x, y))$ is a term containing the variables x and y. For a constant symbol e, we write the corresponding term simply as e instead of $e()$. Some binary function symbols (such as $+$ and $\times$) are written in infix form, with parentheses if necessary, like $(x + y) + z$ instead of $+(+(x, y), z)$.

The main difference between constant symbols and variables is that the latter may be replaced by terms specified with substitutions. A *substitution* is a function $\sigma : V \to T(\Sigma, X)$ that maps every variable to a term. The set of variables $\{x_1, \ldots, x_n\}$ with $\sigma(x_i) = t_i \neq x_i$, $1 \leq i \leq n$, is called the domain of σ. In this case, we may write $\sigma = \{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$. Every substitution σ can be extended to a mapping $\sigma : T(\Sigma, X) \to T(\Sigma, X)$ from terms to terms by introducing a new regulation $\sigma(f(s_1, \ldots, s_n)) = f(\sigma(s_1), \ldots, \sigma(s_n))$. In words, the application of a substitution to a term simultaneously replaces all occurrences of variables by their respective images.

Definition 3: A *rewrite rule* is an ordered pair of terms (l, r) such that l is not a variable and $Var(l) \supseteq Var(r)$. We may write $l \to r$ instead of (l, r). A *term rewriting system* (TRS) is a set of rewrite rules. Note that the rewrite rule can be considered as an equation $l = r$ directed from left to right.

Let $\square$ be a new symbol which does not yet occur in $\Sigma \cup X$. A *context* is a term $C \in T(\Sigma, X \cup \{\square\})$ with a single occurrence of $\square$. For a term s and a context C, $C[s]$ denotes the term obtained by replacing $\square$ in C by s. For any terms $s, t \in T(\Sigma, X)$ and a TRS R, if there exists a rewrite rule $l \to r \in R$, a context C, and a substitution σ such that $s \equiv C[\sigma(l)]$ and $t \equiv C[\sigma(r)]$, we say that s can be *rewritten to* t by a rewrite rule of R and write $s \to_R t$. We call $\to_R$ a reduction relation. A TRS R *terminates* if it allows no infinite rewrite sequences $s_0 \to_R s_1 \to_R \cdots$. In this case, one often says that R is *terminating* or R has the *termination* property. We can prove the termination of term rewriting systems by using the following definition and theorem on reduction orders.

Definition 4: A strict partial order $>$ on $T(\Sigma, X)$ is called a *reduction order,* if it satisfies the following properties.

- closed under context: $s > t$ implies $C[s] > C[t]$, for all contexts C.
- closed under substitution: $s > t$ implies $\sigma(s) > \sigma(t)$ for all substitutions σ.
- well-founded: there are no infinite decreasing sequences $s_0 > s_1 > \cdots$.

Theorem 1: A term rewriting system R terminates if, and only if, there exists a reduction order $>$ that satisfies $l > r$ for all rewrite rules $l \to r$ of R.

3 PARALLELIZATION

3.1 Programming Language Erlang

To implement the termination checker efficiently in a multi-core CPU, we have adopted a programming language named Erlang [7]. Erlang is a general-purpose concurrent programming language run on an efficient runtime system. The sequential subset of Erlang is a functional language, with strict evaluation, single assignment, and dynamic typing. For concurrency it follows the Actor model. It was designed by Ericsson to support distributed, fault-tolerant, soft-real-time, non-stop applications. It supports hot swapping, so that code can be changed without stopping a system. We have selected this language because of the following three characteristics.

Pattern Matching

A term is either a variable or a function symbol followed by zero or more argument terms. To store terms in memory, we often design a recursively-defined tree-like structure, distinguishing between variables and function symbols by using different data types or naming conventions. To conduct term rewriting, we compare the structure of two terms (the pattern and the data) and decide the substitution σ, if it exists, to rewrite the data by a rewrite rule with that pattern in its left-hand side. Erlang has a convenient mechanism, *pattern matching*, which can be used for this purpose. An expression of the form $L = R$ in Erlang means the instruction for matching the value of R with the pattern of L. If they match well, the variables in L will get the corresponding value in R; otherwise there will be an error. For example, if you run in the shell of Erlang the commands $X = 1 + 2$, $Y = X + 3$, $Y = 6$, $X = Y$, you will easily get $X = 3$ and $Y = 6$ from the first three commands. However, the last one will throw a bad-match error, because X is not equal to Y. Pattern matching in Erlang is simple, but when the left-hand side of the equation has a complex structure, it becomes very convenient to valuate all the variables in it. There are two data structures in Erlang we would like to mention. A *tuple* is a structure with a fixed number of data specified in the form $\{x_1,\ldots,x_n\}$ with fixed *n*, while a *list* is a structure with a variable number of data specified in the form $[x_1,\ldots,x_n]$. If you want to get values from a complex tuple like $\{a,[b,c,\{d,e,[f,g]\}]\}$, you only need to do the pattern matching $\{A,[B,C,\{D,E,[F,G]\}]\} = \{a,[b,c,\{d,e,[f,g]\}]\}$ to get the variables in uppercase letters valuated with the data in lowercase letters. Such characteristics make it convenient to deal with TRSs.

Efficient Parallelization

An amazing thing to the users of Erlang is the fact that the program will run n times faster in a n core CPU without any modification. But to achieve this, you must make sure that the program is constructed with processes and there are no interferences and sequential bottlenecks among them. To avoid sequential bottlenecks in the implementation, you can use

the feature named the *process link* in Erlang. After creating a process P_b, you can link it with an existing process P_a for message transfer. A process will send a signal to the linked processes once its task has been completed (or exit with error), and the processes which have received the termination signal also terminate unless they are system processes. A system process can be set at the beginning of the process. This link mechanism is a great help in relieving sequential bottlenecks in the implementation.

Extendability

To communicate with other applications or programs, Erlang can create a process called a *port*. Ports provide your programs with various features to cooperate with external programs. The external programs are run outside the Erlang runtime system. The virtual machine running the Erlang processes copies data through the port to and from the port's driver controlling the external programs. Messages can be sent to a driver through a port by using the same operator, !, used to send messages to regular Erlang processes. Messages sent by drivers to Erlang are also received using the same operator, receive. With this mechanism, your Erlang programs can be easily extended with external programs in a transparent way.

3.2 Lexicographic Path Order

To verify the termination, we use the lexicographic path order (LPO), which is a basic reduction order used in the literature.

Definition 5: Let Σ be a finite signature and $>$ be a strict partial order (called a *precedence*) on Σ. The *lexicographic path order* $>_{lpo}$ on $T(\Sigma, X)$ induced by $>$ is defined as follows:

$s >_{lpo} t$, if and only if

(**LPO1**) s is not a variable and t is a variable that occurs in s, or

(**LPO2**) $s = f(s_1,\ldots,s_m)$, $t = g(t_1,\ldots,t_n)$, and

(**LPO2a**) there exists $i, 1 \le i \le m$, with $s_i \ge_{lpo} t$, or

(**LPO2b**) $f > g$ and $s >_{lpo} t_j$ for all $j, 1 \le j \le n$, or

(**LPO2c**) $f = g$, $s >_{lpo} t_j$ for all $j, 1 \le j \le n$, and

there exists $i, 1 \le i \le m$, such that $s_1 = t_1,\ldots,s_{i-1} = t_{i-1}$ and $s_i >_{lpo} t_i$.

The definition of the LPO is recursive, since in (LPO2a), (LPO2b) and (LPO2c) it refers to the relation $>_{lpo}$ to be defined. Nevertheless, $>_{lpo}$ is well defined, since the definition of $s >_{lpo} t$ only refers to the relation $>_{lpo}$ applied to pairs of terms that are smaller than the pairs s, t. It is proved that $>_{lpo}$ is a reduction order, so the termination of TRS R with the signature Σ is

proved if we can find out a precedence $>$ over Σ such that the LPO $>_{lpo}$ induced by $>$ satisfies $l >_{lpo} r$ for all rewrite rules $l \to r$ of *R*.

3.3 Data Structure

Now we are ready to present the data structure for parallelizing the termination checker. Since only terms and rewrite rules need to be constructed by the data structure, we define their representations in Erlang using the tuple data type as follows:

- A variable v is represented by a tuple $\{v\}$ with a single element.
- A constant c is represented by a tuple $\{c,[]\}$ of a symbol and the empty list.
- A term with the function symbol Fun and its arguments $Arg1, Arg2, \cdots$ is represented by a tuple $\{Fun,[Arg1, Arg2, \cdots]\}$ of a symbol and a non-empty list.
- A rewrite rule $Left \to Right$ is represented by a tuple $\{Left, Right\}$ of two elements where the second one is not a list.

This definition is based on the recursive definition of terms. Note that those four kinds of objects can be clearly distinguished by their syntactical patterns. With those representations, we can store any objects for TRSs in the Erlang environment. For example, we can store the TRS given in Figure 1 in a TRS file as in Figure 2.

$$\begin{cases} not(not(x)) \to x \\ not(or(x,y)) \to and(not(x), not(y)) \\ not(and(x,y)) \to or(not(x), not(y)) \\ and(x, or(y,z)) \to or(and(x,y), and(x,z)) \\ and(or(y,z), x) \to or(and(x,y), and(x,z)) \\ or(or(x,y), z) \to or(x, or(y,z)) \end{cases}$$

Figure 1. An Example of TRS

$$\begin{cases} \{\{not,[\{not,[\{x\}]\}]\}, \\ \quad \{x\}\} \\ \{\{not,[\{or,[\{x\},\{y\}]\}]\}, \\ \quad \{and,[\{not,[\{x\}]\},\{not,[\{y\}]\}]\}\} \\ \{\{not,[\{and,[\{x\},\{y\}]\}]\}, \\ \quad \{or,[\{not,[\{x\}]\},\{not,[\{y\}]\}]\}\} \\ \{\{not,[\{and,[\{x\},\{y\}]\}]\}, \\ \quad \{or,[\{not,[\{x\}]\},\{not,[\{y\}]\}]\}\} \\ \{\{and,[\{x\},\{or,[\{y,\{z\}]\}]\}, \\ \quad \{or,[\{and,[\{x\},\{y\}]\},\{and,[\{x\},\{z\}]\}]\}\} \\ \{\{and,[\{or,[\{y\},\{z\}]\},\{x\}]\}, \\ \quad \{or,[\{and,[\{x\},\{y\}]\},\{and,[\{x\},\{z\}]\}]\}\} \\ \{\{or,[\{or,[\{x\},\{y\}]\},\{z\}]\}, \\ \quad \{or,[\{x\},\{or,[\{y\},\{z\}]\}]\}\} \end{cases}$$

Figure 2. A TRS File

Although the TRS file is difficult for us to read, Erlang can read it easily by using the pattern matching. For example, we only need to match an object with the pattern {_} ("_" means a "wild card" matching with any data) to decide whether it is a variable or not.

Besides the data structures for TRSs, we define the data structure for a precedence $>$ over the function symbols as the list of binary tuples of them. For example, the precedence defined by $f > g$, $g > h$ and $f > h$ is represented as the list $[\{f,g\},\{g,h\},\{f,h\}]$.

Since there is no precedence when the termination checker starts, we initiate it as $I = []$. When we need to add a new element $f > c$, we put $\{f,c\}$ into I, making sure that it preserves the properties required for precedences, without causing no conflict with the current precedence represented as I.

3.4 Parallelization

In this subsection we describe the parallelization architecture for termination verification. Actually, we propose two schemes of parallelization: microlevel and macrolevel.

Microlevel parallelization

First, the architecture for the termination verification of a single TRS is shown in Figure 3.

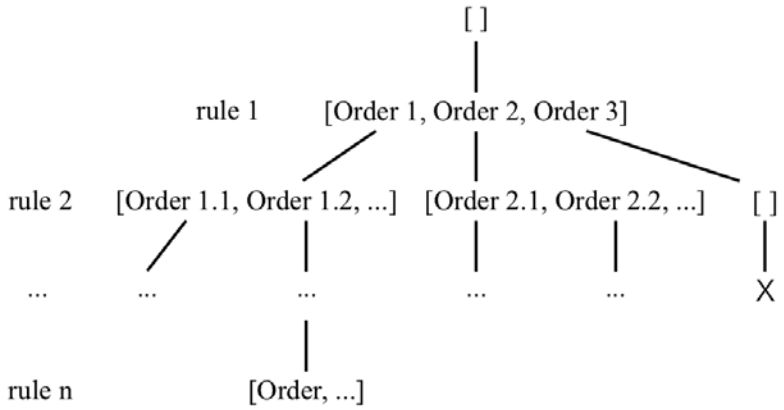


Figure 3. Microlevel parallelization in termination check with Lexicographic Path Orders

If the TRS is empty, the procedure terminates; otherwise it computes the list of the precedences $>^{(i)}$, $i = 1,2,\cdots$, that make the left-hand side of the first rule greater than its right-hand side in $>_{lpo}^{(i)}$. This can be easily computed from the definition of LPO [8]. The procedure then tries to extend each precedence $>^{(i)}$ obtained, so that it can further compute the list of the precedences $>^{(i,j)} \supseteq >^{(i)}$, $j = 1,2,\cdots$, that make the left-hand side of the second rule greater than its right-hand side in $>_{lpo}^{(i,j)}$. The procedure continues this operation until it finds a

precedence $>^{(i,j,\cdots)}$ that makes the left-hand side of the last rule greater than its right-hand side in $>_{lpo}^{(i,j,\cdots)}$ or else it finds that there is no such precedence in any branches. In the figure, a single [] means there is no such precedence. (This should be distinguished from [[]], which means there is an empty precedence in the list.) At each choice point, the procedure creates a parallel process for each precedence just obtained. This procedure can be summarized as follows.

1. Create a supervisor process to monitor the set of created processes by the link feature of Erlang, and create and start a process that executes the step 2 with the rule number $i=1$ and the precedence $p=[]$. The supervisor process waits until a created process returns "terminate" successfully or else it finds there are no created processes, and returns "terminate" in the former case and "failure" in the latter case.
2. Given i and p, if there is no ith rule, then return "terminate"; otherwise compute $P_{list}=\{p_1, p_2, \ldots\}$, which is the list of all the precedences that are extensions of p and make the left-hand side of the ith rule greater than its right-hand side in the induced LPO, and
   - terminate this process, if P_{list} is empty.
   - execute the step 3, if P_{list} is not empty.
3. For each $p_j \in P_{list}$, create and start a process that executes the step 2 with $i+1$ and p_j.

One may notice a subtle synchronization problem in the procedure: if the supervisor process starts before any other children processes, it will return "failure" even before the termination verification is started. To avoid this, we lock the supervisor process until all the possible precedences are obtained and sent to children processes. Besides this, there is also another small synchronization problem, but it is solved by the link and message transfer features of Erlang.

Macrolevel parallelization

Now let us think about two or more TRSs to verify the termination of. Our architecture for such an application is based on the macrolevel viewpoint as shown in Figure 4, where the termination checker should take a stream of TRSs and their identifiers as input. We create an input port in Erlang which receives TRSs from other applications or programs. Each received TRS is assigned to a new process, and its termination will be checked using the procedure described above (designated as lpo in the figure). When the verification is over, the result is sent to the output port to send it to the external program. In Erlang, each process is executed on its independent memory, so there is no interference among the processes in the macrolevel parallelization.

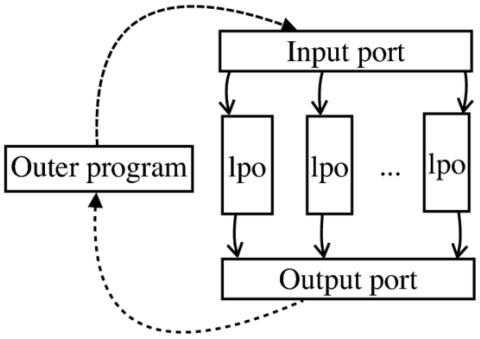


Figure 4. Macrolevel Parallelization in termination check for multiple TRSs

4 EXPERIMENT

In this section, we show and discuss the results of the experiments. In the experiments, we used the standard problems stored in the Termination Problem Data Base [9], which contains 2,125 TRSs to be checked for their termination. The implementation and experiments were performed on a workstation with two AMD Opteron 2.3GHz CPUs with 12 cores. This means we had 24 cores in the workstation.

First we conducted the experiment on all the 2,125 problems. We deliberately eliminated all the IO operation time in the measurement of the computation time so that we only measure the computation time of the termination check. In order to compare the proposed parallelization with the non-parallelization, we also wrote a sequential termination checker in Erlang. We show the result in Figure 5, in which the horizontal axis represents the number of the cores, while the vertical axis represents the computation time in microseconds.

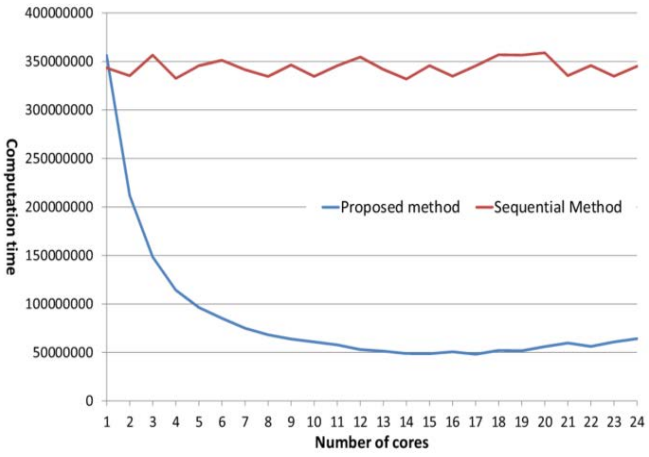


Figure 5. Result of experiment on all the TRSs

With a single core, the computation time of the proposed method was almost the same as the sequential method. With multi-cores, however, the computation time decreased significantly when we increased the number of cores. As for the sequential program, its computation time was basically unchanged, because it creates no parallel processes even when a lot of cores are available. The change in efficiency with the increase of cores can be observed

more clearly in Figure 6, where the vertical axis represents the reciprocal number of the computation time, indicating the speed of computation in terms of the amount of work done in a microsecond.

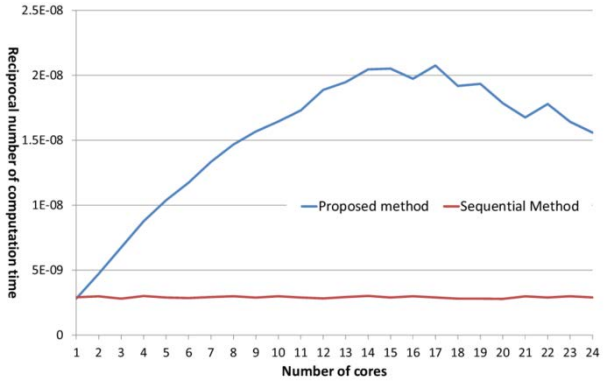


Figure 6. Efficiency result of experiment on all the TRSs

Obviously, the speed of computation increased with the increase in cores, when the number of cores is less than 16. However, when the number of cores exceeded 16, the increase in the efficiency of the proposed method came to the limit and even went down.

In order to figure out the reduction in efficiency, we divided all the TRSs into three groups by the number of rewrite rules they contain. The 46 TRSs which contain 50 or more rules were classified as a group named *Large,* and 300 TRSs with 15 or more but less than 50 rules were named *Medium.* The remaining 1,779 TRSs with less than 15 rules were put into the last group named *Small*.

New experiments were conducted for each of the three groups. First we repeated the same experiment described above, but this time, the program was run for each group as input. In addition, we performed another experiment where only microlevel parallelization was activated for each group. The results are shown in Figure 7, Figure 8 and Figure 9, where "Proposed method" shows the results when both microlevel and macrolevel parallelizations were activated, while "Microlevel Parallel" only activated the microlevel parallelization.

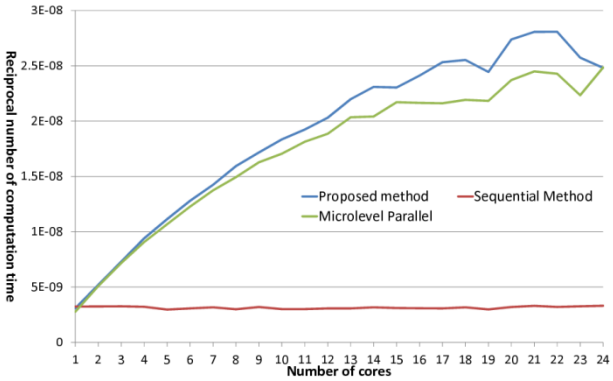


Figure 7. Efficiency result of experiment on Large TRSs

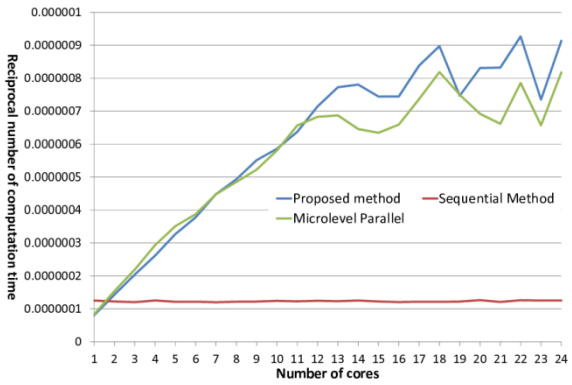


Figure 8. Efficiency result of experiment on Medium TRSs

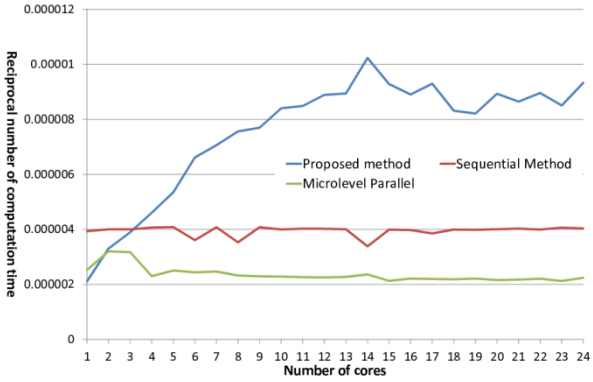


Figure 9. Efficiency result of experiment on Small TRSs

As in the original experiment, the efficiency of the sequential program was almost unchanged with the number of cores, and the efficiency of the proposed method increased with the increase of the number of cores, showing obvious better efficiency than the sequential program. However, a difference from the original experiment can be seen in the figure for the Large group, where the efficiency of the proposed method increased until the number of cores came to 23 rather than the original 16. This is in contrast with the figures for the Medium and Small groups, where the increase in efficiency came to limit when the number of cores became nearly 14.

For the cases of the Large and Medium TRSs, the microlevel parallelization attained almost the same performance as the proposed method, which performs both micro- and macro-level parallelizations. This implies that the microlevel parallelization was effective but the macrolevel one was not effective for those cases. For the case of the Small TRSs, on the other hand, we can see the macrolevel parallelization was very effective and the efficiency of the microlevel parallelization was even worse than the sequential method. The results show that the macrolevel parallelization did not work well for a small number of TRSs, and the microlevel parallelization decreased its efficiency for TRSs with a small number of rewrite rules. From the results, we can say that if the number of TRSs and the number of rewrite rules in those TRSs is large enough, the proposed method is useful and efficient. Fortunately, in practice of

termination checking performed in the inductive theorem proving, we often encounter a large number of complex TRSs, which make the proposed method satisfactory to us.

5 CONCLUSION

In this paper, we have proposed a parallel method of termination checking for term rewriting systems using the lexicographic path order method. The efficiency of the proposed method was shown to be satisfactory for the applications with a large number of complex TRSs generated for termination checking. However, the power of the lexicographic path order is not strong enough to solve a lot of termination problems we encounter. It is a challenging task as a future work to try to parallelize a more powerful termination checking method such as the dependency pair method [6, 10], and finally improve and automate the inductive theorem proving.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number 25330074.

REFERENCES

[1]. Baader, F and Nipkow, T. Term Rewriting and All That, Cambrdge University Press, 1998.

[2]. Reddy, U. Term Rewriting Induction, Proc. of 10th International Conference on Automated Deduction, Lecture Notes in Computer Science, Vol. 449, pp. 162-177, 1990.

[3]. Aoto, T. Rewriting Induction Using Termination Checker, Proceedings of JSSST 24th Annual Conference, 3C-3, 2007.

[4]. Sato, H and Kurihara, M. Muti-Context Rewrting Induction with Termination checkers, IEICE Transactions on Information and Systems, vol. E93-D, no. 5, pp. 942-952, 2010.

[5]. Kurihara, M and Kondo, H. Completion for Multiple Reduction Orderings, Journal of Automated Reasoning, vol. 23, no. 1, pp. 25-42, 1999.

[6]. Arts, T and Giesl, J. Termination of Term Rewriting Using Dependency Pairs, Theoretical Computer Science, vol. 236, no. 1-2, pp. 133-178, 2000.

[7]. Armstrong, J. Programming Erlang: Software for a Concurrent World, Second Edition, Pragmatic Bookshelf, 2013.

[8]. Kurihara, M and Kondo, H. Efficient BDD Encodings for partial order constraints with application to expert systems in software verification, Proceedings of 17th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Lecture Notes in Artificial Intelligence, vol. 3029, pp.827-837, 2004.

[9]. Termination problems data base. [Online] http://termination-portal.org/wiki/TPDB.

[10]. Hirokawa, N and Middeldorp, A. Tyrolean Termination Tool: Techniques and Features, Information and Computation, vol. 205, no. 4, pp. 474-511, 2007.

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 2, ISSUE 4
ISSN: 2054 - 7390

Engineering Analysis and Recognition of Nigerian English: An Insight into Low Resource Languages

Sulyman A. Y. Amuda[1], Hynek Boril[2], Abhijeet Sangwan[2], John H. L. Hansen[2] and Tunji S. Ibiyemi[1]

[1]*Electrical & Electronics Engineering Department, University of Ilorin, Ilorin, Nigeria.*

[2]*Center for Robust Speech Systems, University of Texas, Dallas, USA.*

amudasulyman@gmail.com, hynek@utdallas.edu, sangwan@utdallas.edu, john.hansen@utdallas.edu, ibiyemits@yahoo.com

ABSTRACT

A comparative analysis between Nigerian English (NE) and American English (AE) is presented in this article. The study is aimed at highlighting differences in the speech parameters, and how they influence speech processing and automatic speech recognition (ASR). The UILSpeech corpus of Nigerian-Accented English isolated word recordings, read speech utterances, and video recordings are used as a reference for Nigerian English. The corpus captures the linguistic diversity of Nigeria with data collected from native speakers of Hausa, Igbo, and Yoruba languages. The UILSpeech corpus is intended to provide a unique opportunity for application and expansion of speech processing techniques to a limited resource language dialect. The acoustic-phonetic differences between American English (AE) and Nigerian English (NE) are studied in terms of pronunciation variations, vowel locations in the formant space, mean fundamental frequency, and phone model distances in the acoustic space, as well as through visual speech analysis of the speakers' articulators. A strong impact of the AE–NE acoustic mismatch on ASR is observed. A combination of model adaptation and extension of the AE lexicon for newly established NE pronunciation variants is shown to substantially improve performance of the AE-trained ASR system in the new NE task. This study is a part of the pioneering efforts towards incorporating speech technology in Nigerian English and is intended to provide a development basis for other low resource language dialects and languages.

Index Terms— Nigerian English, Limited Resource Language, Automatic Speech Recognition (ASR)

DOI: 10.14738/tmlai.24.334
Publication Date: 28th August 2014
URL: http://dx.doi.org/10.14738/tmlai.24.334

1 INTRODUCTION

English is spoken by about 130 million people in Nigeria as an official and also a colloquial language. Its unique linguistic characteristics constitute Nigerian English (NE) as a dialect of English. In spite of numerous experimental and instrumental studies of NE, so far a little attention has been paid to building viable speech processing technology for NE or even assessing the dialect-specific speech features from the system engineering perspective. This work presents the first of its kind audio-visual Nigerian English Corpus which consists of 45 hours of speech collected from approximately 530 speakers.

A comparative analysis of Nigerian English alongside with American English (AE) is presented. The two language dialects are closely related in terms of vocabulary but the apparent pronunciation and word choice differences prevent effective application of AE speech technology to NE environments. Hence, our focus is on identifying the major sources of mismatch between the two dialects from the perspective of acoustic signal modeling, evaluation of their impact on automatic speech recognition (ASR) performance, and proposal of an affordable strategy that will allow for a rapid adaptation of an existing ASR engine towards the target domain of the low resource dialect.

NE differs from its counterparts in terms of tones, prosody, phones, and unique lexical patterns that portray the influence of local Nigerian languages [1], [2], [3]. Analyses of speech rhythm and tonal and syllable structures of NE have revealed the tonal nature of the language. Particularly, the pitch employed by NE speakers is lexically significant, contractive, and relative. Additionally, NE is syllable timed and tends to suppress vowel contrast. The observed characteristics in Nigerian English are closely linked to the influence of the major local languages such as Yoruba, Igbo and Hausa. Besides prosodical differences, Nigerian English is also characterized by phonetic differences. The phonetic differences are more obvious when speakers encounter unfamiliar phones that are otherwise absent in their native language [3], often resulting in phone deletion, insertion, or omission. For example, Nigerian English speakers will introduce an unglided vowel structure and unnecessary nasalization of sounds when pronouncing unfamiliar phones while in other cases they may omit phones that are absent in their native language [2, 3].

This study analyzes acoustic-phonetic differences between AE and NE on the level of pronunciation variations, vowel locations in the formant space, mean utterance fundamental frequency, and distances between AE-trained acoustic models and models adapted to NE [4]. It is shown that the AE–NE acoustic mismatch has a strong impact on ASR. In the initial effort towards NE ASR, a combination of model adaptation and extension of an AE lexicon for the newly established NE pronunciation variants is proposed and shown to substantially improve performance of the AE-trained ASR system in the NE task. The results presented here highlight

the challenges brought forth by Nigerian English and are intended to motivate future development of speech systems for limited resource language dialects and languages.

1.1 The Challenges of Nigerian English

Most research works on NE proved that there exist some common phonological properties that can be used to identify NE despite its sub-varieties, hence this work adopts the principles of these properties [5,6,7]. This is based on the concessions of different research works that attempt to explain the accent variability as a direct result of sub-varieties of NE. Some researchers attributed the accent variability to be as a result of different ethnic groups while others believed this to be due to the education background and or the language function or purpose of usage of the language. Another perception for the variability is the influence of the first language on the second language [8] as English language is a second language to all Nigerians.

The common view of this baseline, which is more acceptable by these researchers, is that the standard NE is associated with the minimum of university education and that there is tendency of the speakers' local language to influence the NE in terms of speech rhythm, intonation and accentuation. The phonological aspect of the influence was critically examined by Ulrike Gut (2004) where the consequent effect of the three major languages phonemes on NE is well depicted with respect to British English. The submission is that, the local languages show some varying effects on NE or but there exists some common basis through which the NE clearly differs from the British English. Though Gut did not distinctively define this in terms of phonemes on general terms for NE, or show the common NE phonemic features across the major local languages, all the same it gives a lot of insight into the phonological challenge of the NE. Titi Ufomata (1995) gave an overview of the general Nigerian English phonology as compared with the British English (termed as Received Pronunciation), where pronunciation of some vowels and consonants in NE are undifferentiated and most times confusable or even in some instances totally different from received pronunciation. Some examples given include: "/u/" and "/ʊ/" are both pronounced as "[u]" such that *full* and *fool* are pronounced as [ful]: "/i/" and "/I/" are both pronounced as [i] such that *bead* and *bid* are pronounced as [bid]: "/a/" and "/ae/" are both pronounced as [a] such that *bard* and *bad* are pronounced as [bad]: While /ei/ and /au/ are monophthongized to [e] and [o] respectively, also the fricatives [θ] and [ɗ] are pronounced as [t] and [d] respectively, so as [ʃ] is often pronounced as [ʧ]. Based on these analyses, it was established that NE varied from other English languages in terms of stress, intonation and rhythm.

In spite of the depth of the research on NE, the establishment of NE phonemes is yet to be available. AE therefore, present a good new specimen to apply and expand the speech processing techniques as means of improving ASR systems for limited resources languages based on analysis of certain parameter of speech. The use of AE instead of British English is also justified with the recent steady improvement in relationship between Nigeria and USA. The

influence of the AE on NE is becoming more noticeable in Nigerian academic, social and political circles.

1.2 University of Ilorin Speech (UILSpeech) CORPUS

The UILSpeech corpus was collected as a pioneering effort to form a database for Nigerian English. The speech data in the UILSpeech corpus were exclusively collected at the University of Ilorin campus. Speakers were mostly undergraduate students with an average age of 20 years. The corpus consists of speech from about 300 males and females each. The speaker pool reflects the linguistic diversity of Nigerian English, as most speakers tended to be from 3 dominant linguistic backgrounds in Nigeria, namely, native speakers of Yoruba (South-Western Nigeria), Igbo (South-Eastern Nigeria), and Hausa (Northern Nigeria).

The UILSpeech corpus consists of isolated word recordings as well as continuous read speech data. The isolated word data were collected in a laboratory with the use of a hollow-shaped telephone mouth-piece. The mouth-piece was intended to help reduce speaker-induced variability while ensuring the posture of the speaker. The continuous sentences were recorded with a video camera with the image object distance set between 20 cm to 80 cm. The video data were recorded with a 6.0 mega pixel digital camera, with 640 x 480 resolution and frame rate of 30 frames/sec. Since most data in the corpus were collected in an office/laboratory environment, a low-level background noise is present in the speech utterances. The recorded speech data are

sampled at the rate of 8 KHz for the entire corpus. It is worth mentioning that the speakers were encouraged to speak in a natural manner, and sufficient breaks were given between recording sessions to ensure data quality. Furthermore, the speakers were also subjected to a listener quality evaluation where all speakers in the corpus scored a minimum of 80, 4, and 60 on the Diagnostic Rhyme Test (DTR), Mean Opinion Score (MOS) and Diagnostic Acceptable Measure (DAM), respectively [9, 10].

The isolated word recordings consist of 5 repetitions of 30 different words spoken by 30 different speakers of Nigerian English. A short pause is present between the word repetitions to ensure accurate end-point detection by human annotators and machines alike. The continuous read speech data consist of short utterances spoken by about 500 speakers. The utterances are about 5-15 words long with an average duration of 7.5 seconds. In this manner, the corpus consists of about 15,000 speech utterances in total. Additionally, the continuous speech recordings are also accompanied by a synchronous parallel video recording.

1.3 Dictionaries

Two different dictionaries were used to represent the pronunciations of an American English and Nigerian English. TIMITDICT is used for the AE while, the NE dictionary was developed based on the phonetic transcription of NE by phonetic specialists based on extension of AE

lexicon (this is later referred to as 'NE + NE' lexicon). Consistency and good representation were ensured over the wide range of the data by quantitative corroboration of intra and inter transcription results from the same sets of specialists. The developed NE dictionary only covers the words that were used in this research work whereas the TIMITDICT has over 300, 000 entries.

2 ACOUSTIC-PHONETIC AND SPECTRAL ANALYSIS

In this section, acoustic-phonetic and spectral differences between American English (AE) and Nigerian English (NE) were analyzed along with the impact of the AE–NE acoustic mismatch on ASR. In this study, all NE experiments were conducted on the isolated words portion of the UILSpeech corpus. In particular, 898 utterances from 20 females and 21 males capturing a total of 4490 words formed the NE experimental set. The AE data set were taken from the TIMIT database [11]. TIMIT consists of read speech utterances drawn from 630 speakers of AE (belonging to eight major dialects regions). The TIMIT subset used in the following experiments contains 136 female and 326 male sessions.

Table 1. Example of pronunciation differences in American (AE) and Nigerian (NE) English [4].

Ortographic Transcription	Phonetic Transcription	
	AE	NE
And	/and ae n d/	/ae n t/
Automation	/ao t ah m ey sh ah n/	/ao t ax m ey sh ix n/ /ao t ax m eh sh ix n/
Department	/d ah p aa r t m ah n t/	/d iy p ae t m eh n t/
Electrical	/ax l eh k t r ih k el/	/ax l eh k t r ih k ax l/ /ax l eh t r ih k ax l/ /ax l eh t r ih k ax/
Faculty	/f ae k el t iy/	/f ah k ax l t iy/
Laboratory	/l ae b r ix t ao r iy/	/l ah b ax r ix t r iy/
Numer	/n ah m b axr/	/n uh m b ax/
Zero	/z iy r ow/	/z eh r ax/

2.1 Fundamental Frequency Analysis

The fundamental frequency of speech (F_0) is known to be affected by stress [19, 20], emotions [19, 21], and talking styles [22]. Different languages may exhibit unique F_0 characteristics [23] and the same may be observed also for individual dialects of a language [24]. This motivates the comparative analysis of F_0 in the AE and NE recordings performed in this section. WaveSurfer [16] is used to extract F_0 tracks from the AE and NE utterances. Figure 1 summarizes the mean utterance F_0 values per each dialect ('*AE/NE - All*') followed by gender-specific values ('*AE/NE - Males/Females*'). The error bars represent 95% confidence intervals. It can be seen than in

overall, the NE speakers tend to produce higher-pitched speech compared to AE speakers - the trend being consistent for both genders.

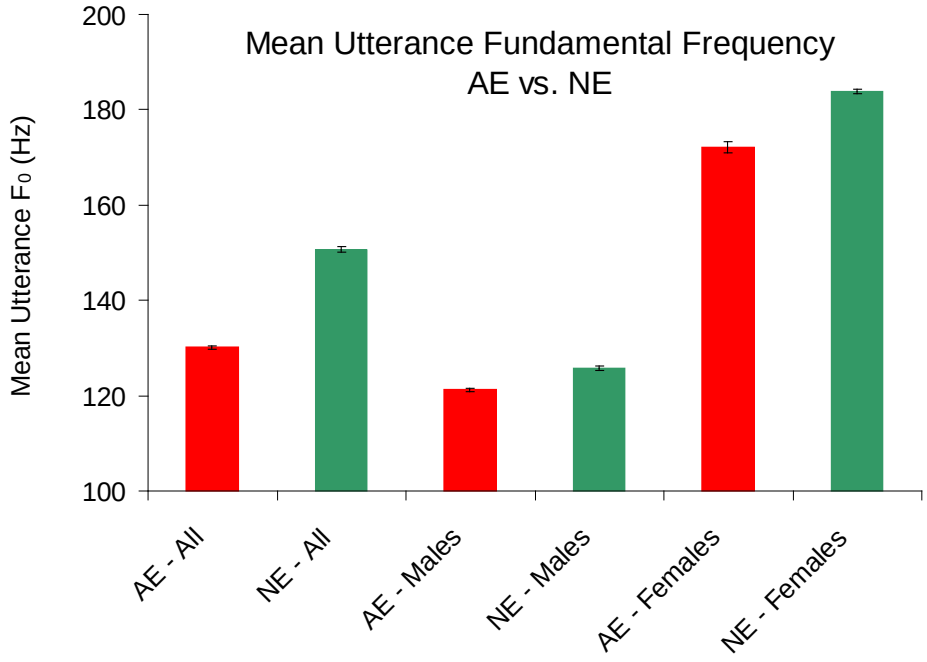
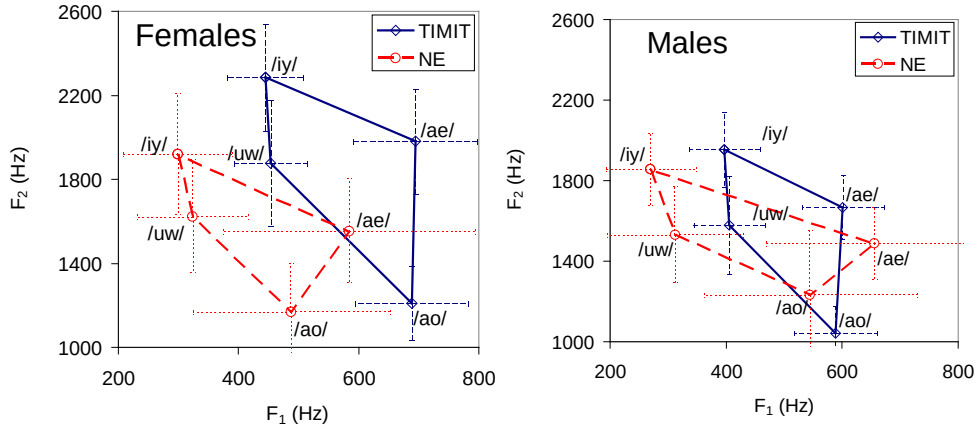


Fig. 1 Comparison of mean utterance fundamental frequency (F_0) in AE and NE recordings.

2.2 Formant Analysis

Past studies of the two languages suggest that there is a phonetic and acoustic mismatch in the AE and NE pronunciations of identical words and phonemes. To better understand the acoustic-phonetic mismatch in the AE and NE data, the locations of vowels in the F_1–F_2 (first and second formant) space are analyzed. Formant frequencies in individual phones are estimated by combining the output of formant tracking (WaveSurfer [16]) and the phone boundaries obtained from forced alignment. In the AE case, the AE lexicon was used in the forced alignment while the NE alignment utilized the 'AE+NE' lexicon. Gender dependent vowel analysis was conducted on the training data sets and the results are shown in Figure 2.

The error bars in the plots represent standard deviations of the F_1, F_2 sample distributions. Compared to native speakers of AE, both F_1 and F_2 vowel coordinates tend to be lower in the NE subjects. This suggests that the NE speakers produce vowels relatively further back and higher as F_1 varies inversely with tongue height and F_2 varies with the posterior-anterior dimension of the vowel articulation [17].



(a) (b)

Fig. 1 Comparison of the phone space of vowels /iy/, /uw/, /ao/, and /ae/ of American English (AE, TIMIT) and Nigerian English (NE) for both (a) female and (b) male [4]

2.3 Inter-HMM Distance Analysis

To further understand the ASR deterioration due to the dialect mismatch, and to get a more detailed insight about the similarities or confusions between the AE and NE phone sets, it is useful to compare the phone spaces of Nigerian and American English in terms of the learned HMM models. For this purpose, we utilize the KL-divergence measurement algorithm proposed in [18] to compute the distances between the baseline AE HMMs and adapted NE HMMs. Fig. 3 shows the KL-divergence between AE and NE vowel and consonant pairs. The articulation characteristics of /ax/ and /ix/ are found to be the closest to each other among AE and NE vowels. On the other hand, the vowels /aw/, /er/, /ay/, /ey/, and /oy/ seem to be the most unfamiliar vowels/diphthongs to NE speakers. The KL-Divergence between every AE and NE HMM pair is shown in Fig. 4. It can be seen that all adapted NE vowel HMMs tend to be closer to the AE /ax/ and /ix/ HMMs. This tendency could be a result of vowel substitutions employed by Nigerian speakers whenever a non-canonical vowel is encountered or when a canonical vowel is encountered in an unfamiliar syllabic position (here, non-canonical vowels refer to the vowels that are native to AE speakers but foreign to NE speakers). For example, the NE vowel /ah/ is close to AE /ah/, /ax/, /eh/, /ih/, /ix/, and /uh/. Here, it is possible that (i) /ah/ in NE is acoustically close to its AE counterpart, as well as (ii) NE speakers tend to substitute the usage of /ah/ with /eh/, /ax/, /ih/, /ix/, and /uh/ in some words. Similar observations can be made for other NE vowels as well, namely, /eh/, /ih/, /iy/, /uh/, and /uw/. For example, as seen in Fig. 4, /ay/ in NE seems to be substituted very frequently by phones /ih/ or /ix/.

Among the NE and AE consonants, /zh/ and /em/ show the largest mismatch, indicating an absence of these phones in NE or a large acoustic mismatch in the speech production. To a lesser degree, fricatives /s/ and /sh/ as well as affricatives /jh/ and /ch/ show a significant mismatch. In general, the acoustic space of the other NE and AE consonants seem to be well matched. However, significant substitutions are indicated among consonants based on the observed distance relationships.

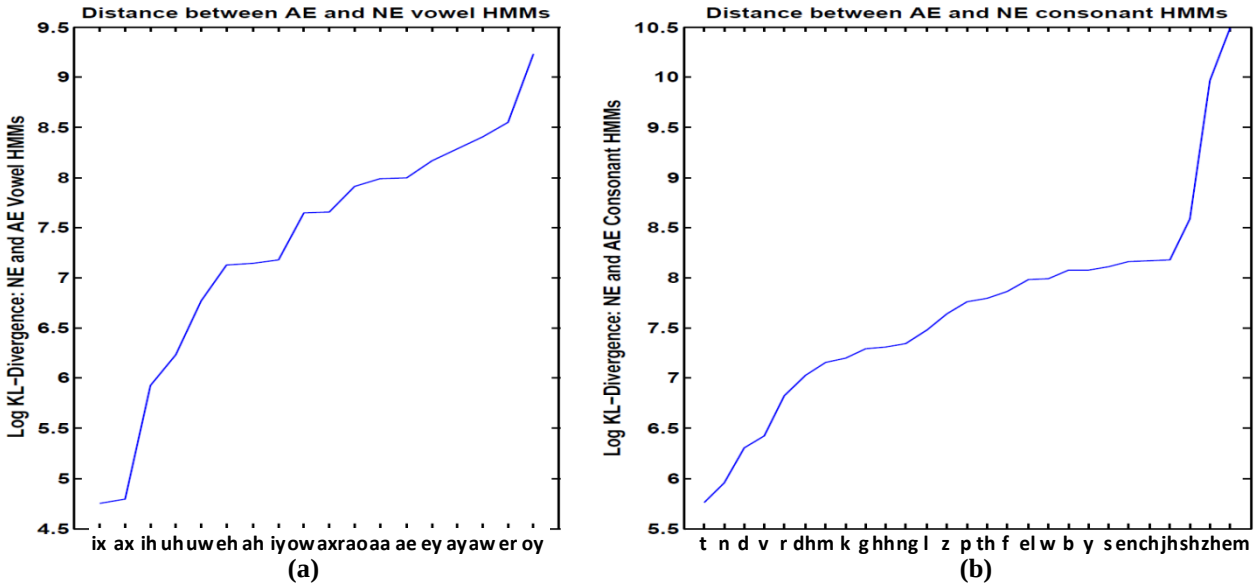


Fig. 3. Comparison of the KL-Divergence for vowels and consonants between (a) Corresponding Nigerian English (NE) and (b) American English (AE) HMMs [4]

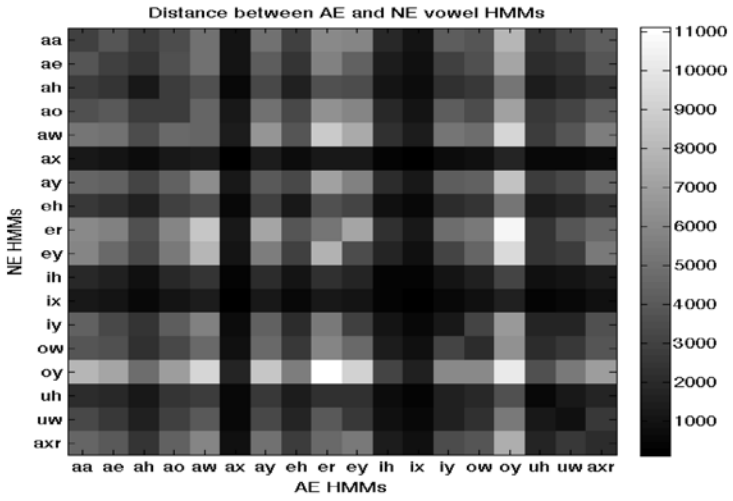


Fig. 4. Histogram-matching for vowels using KL-Divergence between corresponding Nigerian English (NE) and American English (AE) HMMs [4]

2.4 Focused Analysis

Other stimulus parameters of the two languages are studied through a focused analysis of selected word samples. Past research suggests that duration is an essential feature factor in the perception of different accents [11, 12]. Arslan and Hansen used the word final stop closure to carry out analysis of accent classification, with focus on the event before and after the stop consonant in a word. The approach proves to be effective in identifying accent-salient segments in the speech signal. This approach is adopted in our study for words that contain vowels preceding and succeeding a stop consonant, e.g., student, prudent, ardent, apart, etc. Our analysis suggests that in such words the NE speakers tend to spend longer time (put more emphasis) on the vowel after the consonant compared to AE speakers. Example spectrograms of the word 'student' from an NE speaker and an AE speaker are shown in Figure 5 (a) and (b), respectively.

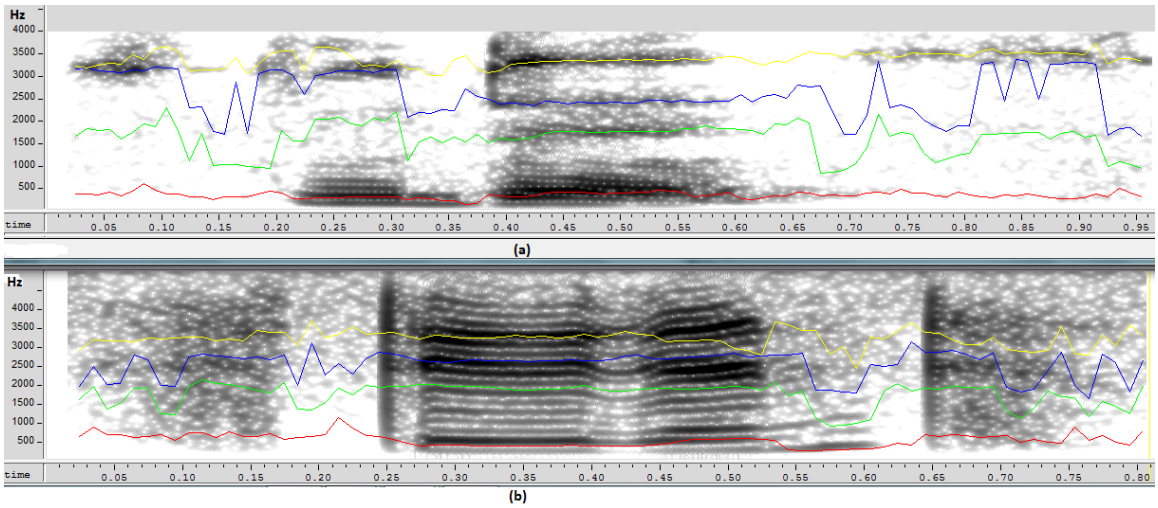


Figure 4. The spectrogram of the word "student" by (a) Nigerian English speaker and (b) American English speaker

3 ASR PERFORMANCE ANALYSIS

In spite of the above analysis, our focus is on rapid migration of an existing speech recognizer trained on American English (AE) to recognize Nigerian English (NE). This is a challenging task as AE and NE differ drastically along a number of critical speech parameters such as phonetic space, intonation patterns, and stress patterns. Considering this, the aim is to primarily mitigate the differences in the phonetic space by employing a two-pronged strategy: (i) developing a Nigerian English lexicon, and (ii) using a popular maximum-a-posteriori (MAP) model-adaptation technique to compensate for the acoustic phoneme pronunciation mismatch in the phone space [17].

3.1 ASR System Baseline

Detailed descriptions of the laboratory setup can be found in [4]. The performance of the baseline ASR system trained on the AE set and utilizing a TIMIT AE pronunciation lexicon is

shown in the second and third row of Table 2 for the complete AE set, denoted 'Devel+Test', and for the subset of AE comprising 1490 words from 9 speakers, denoted 'Test'. It can be seen that despite the simplicity of the small vocabulary task, the performance is very low, reaching approximately 50% word error rate (WER). It is believed that two major factors contribute to the poor performance: (i) the phonetic mismatch in the AE vs. NE pronunciations of the identical words, and (ii) the acoustic mismatch in the pronunciation of the identical phonemes in AE vs. NE.

In order to address the first factor, two trained phoneticians were asked to listen to a portion of the NE utterances and write down the most representative phonetic transcriptions of the 30 vocabulary words (see an example of AE–NE pronunciation differences in Table 1 as observed for TIMIT vs. UISpeech corpora). Subsequently, these transcriptions were used to extend the AE lexicon, yielding a lexicon denoted 'AE+NE'. As shown in rows 4 and 5 in Table 2, employing the extended lexicon helps to reduce WER by 2.5–3% absolute.

To address the phoneme pronunciation mismatch between AE trained acoustic models and NE test data, the acoustic models were adapted to the development ('Devel') set (1355 utterances from 32 female and male speakers who are distinct from the 'Test' set) using the MAP adaptation. First, forced alignment was performed on the development set given the known utterance transcriptions, yielding an estimation of the phone boundaries. Second, multiple MAP adaptation passes were performed. It was observed that 5 passes yielded reasonably adapted speaker-independent models (rows 6 and 7 in Table 2). Note that utilizing the combined 'AE+NE' lexicon in the adaptation process further reduces WER by 5.5% compared to using only the AE lexicon. Finally, an adaptation scheme where phone boundaries were re-estimated in every MAP adaptation iteration using the updated models was also evaluated (see the last row of Table 2). It can be seen that multiple re-alignments with the updated models do not significantly contribute to model refinement. When employing both lexicon extension and model adaptation, the overall absolute WER reduction over the baseline reaches 37%. Table 3 details the impact of the MAP adaptation on recognition performance when applied to a subset of phone models versus all models. It can be seen that adapting only consonant models (penultimate row of Table 3) has more substantial impact than adapting only vowel models - 20.3% absolute WER reduction versus 3.4% over the baseline unadapted models. However, adapting all models brings a further 14.3% WER reduction compared to adapting only consonants.

Table 2. ASR Performance of isolated word recognition part of UILSpeech Corpus. Test – re-alignments are performed every iteration [4].

Training	Lexicon	Set	WER (%)
No MAP	AE	Devel + Test	49.3
		Test	51.0
	AE + NE	Devel + Test	46.3
		Test	48.5
MAP	AE	Test	19.5
	AE + NE	Test	14.0
		Test*	13.9

It is noted that the improvements due to MAP adaptation may also be partly due to model adaptation to the acoustic environment of UILSpeech.

Table 3. ASR Performance of isolated word recognition on UILSpeech Corpus. The impact of MAP adaptation when applied to selected groups of phone models.

Adapted Phone Models	Open Test Set WER (%)
None	48.5
el, iy, ih, eh, ey, ae, aa, aw, ay, ah, ao, oy,ow, uh, uw, er, ax, ix, axr	45.1
b, d, g, p, t, k, m, n, ng, em, en, s, sh, z, zh, f, th, v, dh, jh, ch, l, r, w, y, hh	28.2
All	13.9

4 AUDIO-VISUAL ANALYSIS

Finally, we conduct an informal audio-visual analysis of the AE and NE speaker sessions. The goal is to relate the acoustic-phonetic properties of speech production with articulatory/facial movements. In this analysis, we study video recordings acquired while the subjects were reading three different sentences with three repetitions. The video samples are analyzed frame by frame using AVS Video Editor 4 [25] with a specific focus on the phonemes /dh/, /th/, /d/, /t/, /r/, /f/ and /v/ which are often confusable to Nigerian speakers.

Analysis of the the facial muscles and jaw positions in the video transcriptions reveals frequent substitution of the voiced flap in /dh/ for /d/ by most of the Nigerian speakers (see Figure 6). On the other hand, the expected substitution of /f/ for /v/ could not be ascertained. The two phonemes can be distinguished by a different mouth shape (lip height and width), however, the patterns here are strongly speaker dependent and the visual distinction is complicated by the fact that these phonemes are produced with most of the articulators covered by the lips. The

lip shape patterns for /t/ and /r/ were found strongly dependent on their position in the word (coarticulation effects) and unique to each speaker; that is, the lip shape pattern when producing the same utterance strongly varied across the speakers.

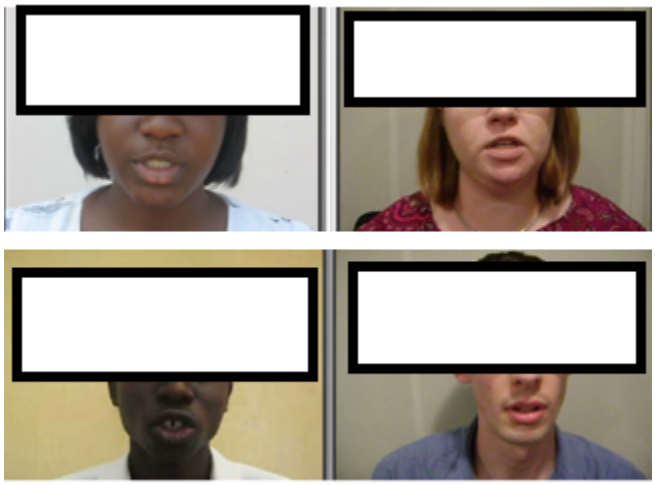


Fig. 6. Comparison of the lip shapes and jaw positions for the phoneme /dh/ production by both genders of Nigerian English (NE) and American English (AE) speakers

5 CONCLUSIONS

The data presented in this article consists of simultaneous speech audio and video tracks that capture isolated and read speech utterances. The corpus provides a unique opportunity for building a variety of speech systems such as speech/speaker recognition and dialect/accent identification for Nigerian English. Analysis of the American English and Nigerian English utterances on the lexical level and in terms of acoustic model distances, mean utterance fundamental frequency, and vowel location in F1–F2 space confirms substantial differences in American English and Nigerian English. Such differences cause a significant deterioration of American English-trained ASR when exposed to Nigerian English. A simple scheme that combines extended American English lexicon for Nigerian English pronunciation variants and multi-pass acoustic model adaptation showed a reduction of recognition errors by 37% absolute WER. These encouraging results suggest that such an approach may represent a viable ASR path also for other low resource dialects with limited availability of speech data and phonetic information. The results also show that while improved lexicon pronunciation is beneficial, corresponding advancement in acoustic modeling for the new language dialect domain is necessary to reach substantial performance gains. The audio-visual analysis of the lip shape patterns during speech production revealed strong speaker dependency for certain phonemes. Visual features extracted from lip shape patterns of these phonemes could be beneficial to speaker authentication applications.

ACKNOWLEDGMENTS

The authors appreciate the financial and moral support from the Fulbright foundation of IIE, U.S.A. and the Center for Robust Speech System (CRSS), Erik Jonsson School of Electrical and Computer Science, The University of Texas at Dallas. We also wish to thank all individuals that contributed to the data collections both in Nigeria and U.S. The portion of the study conducted in CRSS was funded partially by the University of Texas at Dallas from the Distinguished University Chair in Telecommunications Engineering held by J.H.L. Hansen.

REFERENCES

[1] U. Gut and J.-T. Milde, "The prosody of Nigerian English," in *SP-2002*, 2002, pp. 367–370.

[2] C. T. Hodge, "Yoruba: Basic course," ED – 010 – 462 Report NDEA – VI – 375, US Foreign Service Institute, 1963.

[3] A. A. Fakoya, *Nigerian English: A Morpholecta Classification*, Ph.D. thesis, Lagos State University, 2007.

[4] S. Amuda, Boril, H., Sangwan, A. and Hansen, J. H. L. (2010). "Limited Resource Speech Recognition for Nigerian English." *Proc. of IEEE ICASSP'10*, 5090-5093.

[5] M. Jibril, "Phonological Variation in Nigerian English", Ph.D Thesis at University of Lancaster 1986

[6] T. T. Ajani "Is There Indeed A 'Nigerian English'?" *Journal of Humanities & Social Sciences*, 1(1), 2007.

[7] T. Ufomata "Setting Priorities in Teaching English Pronunciation in ESL Contexts", Seminar presentation as a British Academy Visiting Fellow at University College London, 1996.

[8] A. Bamgbose, "Language in Contact: Yoruba and English in Nigeria", *Education and Development*, 2(1), pp. 329-341, 1982.

[9] W. Voiers, I. Dynastat, and T. Austin, "Diagnostic Acceptability Measure for Speech Communication System," in *Proc. of IEEE ICASSP*, vol. 2, pp. 204–207, 1977.

[10] M. A. Koler, "A Comparison of the New 2400 bps MELP Federal Standard with other Standard Coders," in *Proc. of IEEE ICASSP*, 1997.

[11] L. M, Arslan and J. H. L. Hansen, "Language Accent Classification in American English", *Speech Communication*, vol. 18, pp. 353-367, ELSEVIER, 1996.

[12] L. M, Arslan and J. H. L. Hansen, "A Study of Temporal Features Frequency Characteristics in American English Foreign Accent", *Journal of Acoustical Society of America*, vol. 201(1), pp. 28-40, July, 1997.

[13] J. S. Garofolo, L. F. Lamel, J. G. Fisher,W.M. andFiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, *TIMIT Acoustic-Phonetic Continuous Speech Corpus*, LDC93S1, 1993.

[14] J.-L. Gauvain and Chin-Hui Lee, "Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains," *IEEE Transactions on Speech & Audio Processing*, 2(2), pp. 291–298, 1994.

[15] S. B. Davis and P. Mermelstein, "Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 28(4), pp. 357–366, 1980.

[16] K. Sjolander and J. Beskow, "WaveSurfer – An Open Source Speech Tool," in *Proc. of ICSLP'00*, Beijing, China, 2000, vol. 4, pp. 464–467.

[17] R. D. Kent and C. Read, *The Acoustic Analysis of Speech*, Whurr Publishers, San Diego, 1992.

[18] J. Silva and S. Narayanan, "Average Divergence Distance as a Statistical Discrimination Measure for Hidden Markov Models," *IEEE Transactions on Audio, Speech, and Language Processing*, 14(3), pp. 890–906, 2006.

[19] J. H. L. Hansen, "Analysis and Compensation of Speech Under Stress and Noise for Environmental Robustness in Speech Recognition," *Speech Communication*, 20(1-2), pp. 151–173, 1996.

[20] J. H. L. Hansen, E. Ruzanski, H. Boril, J. Meyerhoff, "TEO-Based Speaker Stress Assessment Using Hybrid Classification and Tracking Schemes," *International Journal of Speech Technology*, Springer, June 2012, DOI 10.1007/s10772-012-9165-1.

[21] T. Hasan, H. Boril, A. Sangwan, J. H. L. Hansen, "Multi-Modal Highlight Generation for Sports Videos Using an Information-Theoretic Excitability Measure," *EURASIP Journal on Advances in Signal Processing*, 2013:173, 2013.

[22] H. Boril, J. H. L. Hansen, "Unsupervised Equalization of Lombard Effect for Speech Recognition in Noisy Adverse Environments," *IEEE Transactions on Audio, Speech, and Language Processing*, 18(6), 1379-1393, 2010.

[23] H. Boril, Q. Zhang, A. Ziaei, J. H. L. Hansen, D. Xu, J. Gilkerson, J. A. Richards, Y. Zhang, X. Xu, H. Mao, L. Xiao, F. Jiang, "Automatic Assessment of Language Background in Toddlers Through Phonotactic and Pitch Pattern Modeling of Short Vocalizations," *accepted to Workshop on Child Computer Interaction (WOCCI)*, September, Singapore, 2014.

[24] M. Mehrabani, H. Boril, J. H. L. Hansen, "Dialect Distance Assessment Method Based on Comparison of Pitch Pattern Statistical Models," *in Proc. of IEEE ICASSP'10*, 5158-5161, Dallas, TX, 2010.

[25] Link: http://www.avs4you.com (accessed on Aug 20, 2014).

TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
SOCIETY FOR SCIENCE AND EDUCATION UNITED KINGDOM
VOLUME 2, ISSUE 4
ISSN: 2054 - 7390

Anisotropy of Ferromagnetic Materials

Fae`q Radwan
Faculty of Engineering, Near East University,TRNC - Lefkosa Mersin-10, TURKEY

ABSTRACT

The decomposition of elastic constant tensor into its irreducible parts is given. The norm of elastic constant tensor and the norms of the irreducible parts of the elastic constants of ferromagnetic materials (the elements Fe, Ni and Co and some of their alloys) are calculated. The relation of the scalar parts norm and the other parts norms and the anisotropy of the ferromagnetic materials are presented. The norm ratios are used to study anisotropy of ferromagnetic materials and the relationship of their structural properties and other properties with their anisotropy are given.

Keywords: Ferromagnetic, Norm, Anisotropy, and Elastic constants.

1 INTRODUCTION

The decomposition procedure and the decomposition of elastic constant tensor is given in [1,2,3,4,5,6], also the definition of norm concept and the norm ratios and the relationship between the anisotropy and the norm ratios are given in [3,4,5,6]. As the ratio N_s/N becomes close to one the material becomes more isotropic, and as the ratio N_n/N becomes close to one the material becomes more anisotropic as explained in [3,4,5,6].

2 CALCULATIONS

Let us consider the irreducible decompositions of the elastic constant tensor in the following crystals

Table 1, Elastic Constant (GPa), [7]

Element, Cubic System	c_{11}	c_{44}	c_{12}
$\beta - Cobalt, Co^{a)}$	260	110	160
$\nu(n=3)$	35	36	1
Iron, Fe	230	117	135
$\nu(n=10)$	2	1	3
Nickel, Ni, Zero field	247	122	153

DOI: 10.14738/tmlai.24.301
Publication Date: 28th August 2014
URL: http://dx.doi.org/10.14738/tmlai.24.301

$\nu(n=4)$	2	2	3
Nickel, Ni, Saturation field $\nu(n=4)$	249 1	124 1	152 3

Table 2, Elastic Constants (GPa) [7]

Alloy	c_{11}	c_{44}	c_{12}
Cobalt – Iron, Co-Fe at % of Fe 6	234.0	125.9	158.9
8	232.7	124.8	159.8
12	228.7	122.9	160.0
14	226.5	121.3	160.4
Iron-Nickel, Fe-Ni at % Ni 27.2	160.8	116.0	95.8
29.0	152.6	113.1	91.6
33.3	133.3	105.9	85.7
30.4	140.4	112.1	84.0
32.1	136.2	108.6	85.2
34.2	135.6	104.2	91.0
36.5	150.7	102.0	107.7
38.8	159.2	102.4	116.2
41.3	171.3	102.9	126.1
50.2	205.3	107.5	145.9
73	230.4	119.2	144.4
At % Ni, 4.2 K 35	157.3	100.6	123.5
59.6	228.3	117.6	150.1
77.6	247.6	127.7	151.2
89.2	254.6	130.0	152.8
100	261.4	130.9	154.8

By using table 1and table 2, and the decomposition of the elastic constant tensor, we calculated the norms and the norm ratios as in table 3 and in table 4.

Table 3, the norms and norm ratios

Element	N_s	N_d	N_n	N	N_s / N	N_d / N	N_n / N
$\beta - Cobalt, Co^{a)}$ $\nu(n=3)$	646.3405	0	109.9818	655.631	0.98583	0	0.16775
Iron, Fe $\nu(n=10)$	580.9673	0	127.3956	594.7711	0.976791	0	0.214193
Nickel, Ni, Zero field $\nu(n=4)$	631.5956	0	137.4773	646.3846	0.97712	0	0.212687
Nickel, Ni, Saturation field $\nu(n=4)$	634.5013	0	138.3938	649.4187	0.97703	0	0.213104

Table 4, the norms and norm ratios

Alloy	N_s	N_d	N_n	N	N_s / N	N_d / N	N_n / N
Cobalt – Iron, Co-Fe at % of Fe 6	628.2480	0	161.9482	648.7856	0.968345	0	0.249617
8	626.9524	0	161.9482	647.5311	0.96822	0	0.250101
12	620.6867	0	162.3148	641.5591	0.967466	0	0.253001
14	617.0188	0	161.6733	637.8483	0.967344	0	0.253467
Iron-Nickel, Fe-Ni at % Ni 27.2	446.3587	0	153.0580	471.8717	0.945932	0	0.324364
29.0	428.0973	0	151.4083	454.0834	0.942772	0	0.333437
30.4	403.6885	0	153.7912	431.9909	0.934484	0	0.356006
32.1	395.5679	0	152.3248	423.883	0.933201	0	0.359356
33.3	389.3039	0	150.4918	417.379	0.932735	0	0.360564

34.2	396.2451	0	150.1252	423.7307	0.935134	0	0.354294
36.5	433.1531	0	147.5589	457.5973	0.946582	0	0.322465
38.8	455.3155	0	148.2921	478.8557	0.950841	0	0.30968
41.3	484.2108	0	147.1923	506.0887	0.956771	0	0.290843
50.2	557.9259	0	142.6098	575.8636	0.968851	0	0.247645
73	596.7849	0	139.6769	612.9126	0.973687	0	0.227890
At % Ni, 4.2 K 35	461.5350	0	153.4246	486.3678	0.948942	0	0.31545
59.6	600.8719	0	143.8929	617.861	0.972503	0	0.232889
77.6	635.3463	0	145.7259	651.8443	0.97469	0	0.223559
89.2	648.2635	0	144.9927	664.2804	0.975888	0	0.21827
100	660.1009	0	142.2431	675.2528	0.977561	0	0.210652

3 CONCLUSION

- From table (3), considering the ratio $\frac{N_s}{N}$ we can say that Copper, (first ionization energy is 745KJ/mole) is more isotropic than Nickel, (first ionization energy is 577.9KJ/mole), and considering the value of N which is more high in the case of Nickel, so can say that Nickel elastically is stronger than Copper, and Nickel with saturation field is more anisotropic and elastically is stronger than Nickel with zero field.
- From table (4) considering the ratio $\frac{N_s}{N}$ we can say that in the Alloy Cu-Al as the percentage of Ni increases the anisotropy of the alloy increases, and considering the value of N which is increasing as the percentage of Ni increases, so can say that the alloy becomes elastically strongest.

REFERENCES

[1]. Jerphagnon, J., Chemla, D. S., and Bonnevile, R., (1978), "The Description of Condensed Matter Using Irreducible Tensors", Adv. Phys, 11, p1003-1017.

[2]. Radwan, Fae'q A. A. (1999), " Norm Ratios and Anisotropy Degree", The First International Conference in Mathematical Sciences, Applied Mathematics, ICM99, November 1999, United Arab Emirates University, Al Ain, United Arab Emirates. Pak. J. Appl. Sci..Vol. 1, (3): 301-304, 2001.

[3]. F. A. A. Radwan, 'Scalar Irreducible Parts of Sixth Rank Tensor', Arab Gulf Journal of Scientific Research, 19 (3), Pp163-166, (2001).

[4]. Fae`q A. A. Radwan, ' Some Properties of Copper - Gold and Silver - Gold Alloys at Different % of Gold', Journal: Lecture Notes in Engineering and Computer Science Year: 2011, Vol 2189, Issue: 1, pp1221-1224.

[5]. Fae`q A. A. Radwan, ' Isotropy of Some Types of Bones`, International Journal of Engineering Science and Technology (IJEST), Vol. 3 No. 4 April 2011: 3382-3386.

[6]. Fae'q A. A. Radwan, 'Some properties of copper-nickel at different % of Ni', Pelagia Research Library, Der Chemica Sinica, 2014, 5(1):34-38.

[7]. Landolt-Börnstein, Group III, "Crystal and Solid State Physics", Volume, 11, Springer-Verlag.