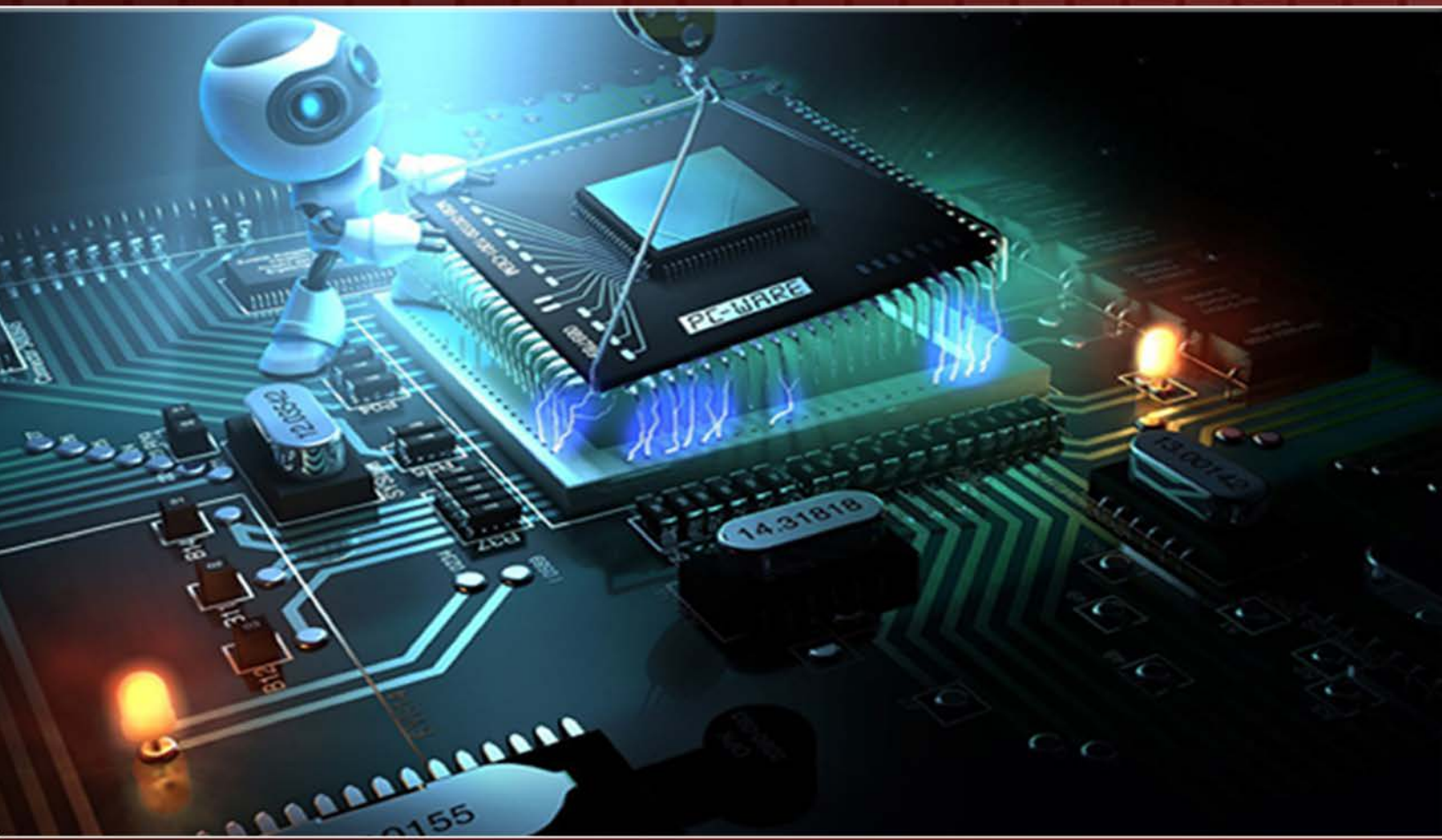
ISSN: 2054-7390

Volume 3 Issue 2, April 2015

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.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
VOLUME 3, ISSUE 2
ISSN 2054-7390

An Objective Approach to Schizophrenia Recognition Utilizing an Adaptive Neuro-Fuzzy Inference (ANFIS) Model

Amadin, F. I[1] and Obi, J.C.[2]
Department Of Computer Science, University Of Benin, Benin City. Nigeria
[1]frankamadin@uniben.edu; [2]tripplejo2k2@yahoo.com

ABSTRACT

Schizophrenia is a brain disorder that distorts the way a person thinks, acts, expresses emotions, perceives reality, and relates to others. A systematic approach and an overview perception has been carried out over the years by different researchers, but none has sufficiently introduced and Adaptive Neuro-Fuzzy Inference System (ANFIS) approach for these prediction which has served as the focal aim of this research paper using available parameters and datasets. Matric Laboratory (MATLAB) 7.0 served as the tool of implementation highlighting various views. The ANFIS training was successful completed at epoch 34, and had an error of 2.47367e-005 and the test error which was generated by the ANFIS was 0.0002511. The training was accomplished using a constant membership function type at an error tolerance at 0.05.

Keyword: Schizophrenia, ANFIS, Membership Function, Training and Testing Errors

1 Introduction

Schizoaffective disorder is a serious mental illness that has features of two different conditions, schizophrenia, and an affective (mood) disorder that may be diagnosed as either major depression or bipolar disorder (WebMD, 2014).

Schizophrenia is a brain disorder that distorts the way a person thinks, acts, expresses emotions, perceives reality, and relates to others. Depression is an illness that is marked by feelings of sadness, worthlessness, or hopelessness, as well as problems concentrating and remembering details (WebMD, 2014). Bipolar disorder is characterized by cycling mood changes, including severe highs (mania) and lows (depression).

Schizoaffective disorder is a lifelong illness that can impact all areas of daily living, including work or school, social contacts, and relationships. Most people with this illness have periodic episodes, called relapses, when their symptoms surface. While there is no cure for schizoaffective disorder, symptoms often can be controlled with proper treatment (WebMD, 2014).

While the exact cause of schizoaffective disorder is not known, researchers believe that genetic, biochemical, and environmental factors are involved (Help Guide, 2014 WebMD, 2014 and Right Diagnosis, 2014:

- **Genetics (heredity)**: A tendency to develop schizoaffective disorder may be passed on from parents to their children.

DOI: 10.14738/tmlai.32.1035
Publication Date: 7th April, 2015
URL: http://dx.doi.org/10.14738/tmlai.32.1035

- **Brain chemistry:** People with schizophrenia and mood disorders may have abnormalities in the functioning of brain circuits that regulate mood and thinking.
- **Environmental factors**: Theories suggest that certain environmental factors such as a viral infection, poor social interactions or highly stressful situations may trigger schizoaffective disorder in people who have inherited a tendency to develop the disorder. However, the relationships between biological and environmental factors that may lead to schizoaffective disorder are not well understood.

The positive symptoms of Schizophrenia are delusion, hallucination, disorganised speech and thinking affect and catatonic behaviour the negative while the negative symptoms associated with schizophrenia occur as a result of degeneration of everyday activity which might include Flattened Affect, Alogia and Avolition also known as loss of motivation from the medical point of view, the person might show lack or disinterest in socializing (Mayoclinic, 2014 and webMd, 2014). This is as a result of degeneration in catatonic behaviour. Both positive and negative symptoms of Schizophrenia has their functionalities; if the patients displays only positive symptoms then the patient might be suffering from acute schizophrenia whereas the chronic schizophrenia occur when patients displays signs of negative symptoms (HelpGuide, 2014).

According to the Diagnostic and Statistical Manual of Mental Disorder IV (DSMMD-IV) for a patient to be diagnosed with schizophrenia both positive and negative characteristic signs and symptoms (both positive and negative) must be present for a significant period (a month) with the symptoms delusion and hallucination persisting for at least 6 months. Schizophrenia can be classified under five categories they are (MedicineNet, 2014, RightDiagnosis, 2014 and WebMD, 2014):

- Paranoid Schizophrenia: The patient is obsessed with someone following him/her, spicing on him/her or tricking him/her. It might even involve symptoms like hallucinations but muddle behaviour of loss of speech is not evident here.
- Disorganized Schizophrenia: this form of schizophrenia involves positive symptoms like disorganized speech and behaviour which might also show loss of self-expression.
- Catatonic Schizophrenia: Patient suffering from catatonic schizophrenia might exhibit severe bewilderment in behaviour
- Undifferentiated Schizophrenia: A patient suffering from undifferentiated schizophrenia might show signs of delusions, hallucinations, disorganized speech or behaviour, catatonic behaviour or negative symptoms.
- Residual Schizophrenia: A patients suffering from residual schizophrenia must show signs of only negative symptoms.

2 Review of Related Literature

Several research works has focused on Schizoprehnia,

Dwight et al., (2010), carried out a research on comparison of cognitive structure in schizophrenia patients and healthy controls using confirmatory factor analysis based on the underlining evidence that cognitive task performance breaks down into the same broad domains in schizophrenia. In reaching a conclusive boundary, a confirmatory factor analysis (CFA) to compare the latent structure of a broad

neuropsychological battery in schizophrenia patients (n= 148) and healthy controls (n= 157) were utilized. Conclusively it was agreed that CFA possesses higher efficiency.

Tara et al., (2013) focused on suicide prevention in schizophrenia spectrum disorders and psychosis embedding a systematic approach using Cochrane, PubMed and PsycINFO databases dataset as methodology. Conclusively, it was agreed that Psychosocial interventions may be effective in reducing suicidal behaviour in patients with schizophrenia spectrum disorders and psychosis, although the additional benefit of these interventions above that contributed by a control condition or treatment-as-usual is not clear.

Other research work includes (Víctor and Manuel, 2003; Steffen and Todd, 2007, Olivier and Marc, 2005).

From the exhaustive literature review, an Objective approach has rear been adopted for the diagnosis of schizophrenia, based on this demerit; An Objective Approach to Schizophrenia Recognition Utilizing an Adaptive Neuro-Fuzzy Inference (ANFIS) Model was inspired.

3 Methodology: Adaptive Neuro Fuzzy Inference System (ANFIS) Architecture

The Adaptive Neuro Fuzzy System is one of the several types of Neuro-fuzzy system that combines both the learning capabilities of the Neuro fuzzy system and the explanatory power of the fuzzy inference system. It has 6 layers and each layer comprises of neurons that performs specific function. The fuzzy inference system can be built using the mamandi or the sugeno-tagaki model. The ANFIS uses the sugeno-tagaki model in building the fuzzy inference system.

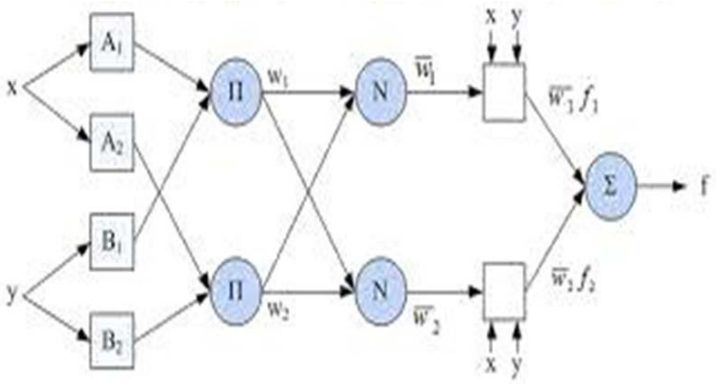


Figure 2: Adaptive Neuro-Fuzzy Inference System (ANFIS) Model for Schizophrenia Application

a) Layer 1: (Input Layer): This is the first layer it is called the input layer.
b) Layer 2: (Fuzzification layer): The second layer of the Adaptive Neuro Fuzzy Inference System (ANFIS) model is called fuzzification layer. It could also be called the membership function layer. In this layer the input coming in from the input layer is mapped to fuzzy set using the bell membership function.
c) Layer 3: (Rule Layer): The third layer in the ANFIS is called the rule layer. The rules in this layer are built using the sugeno rule fuzzy model. Each neuron in this layer receives the input from the fuzzification layer and computes the output.

$$O_{2,i} = w_i = \mu_{A_i}(x) \times \mu_{B_i}(x) \qquad i = 1,\ 2$$

d) Layer 4 (Normalization layer): The fourth layer of the ANFIS is called the normalization layer. The neurons in this layer receive inputs from the rule layer and calculate the normalization firing strength. This is sent to the fifth layer.

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2} \qquad i = 1,\ 2$$

a) Layer 5 (Defuzzification layer): The fifth layer in the Adaptive-Neuro Fuzzy System is called the defuzzification layer. The neurons in this layer receive it's input from the fourth layer.

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i)$$

b) Layer 6 (Output Layer): the sixth layer of the Adaptive Neuro Fuzzy Inference System is called the output layer. It give the overall out of the Adaptive Neuro Fuzzy System. It contains a single neuron that performs summation of all the incoming inputs from the fifth layer.

$$O_{5,1} = overall\ output = \sum_i \overline{w}_i f_i$$

4 Stimulations and Results

The dataset used for the computer stimulation was retrieved from eight-two case sample was randomly selected from the sample and it contained 76% schizophrenic cases and 24% free cases. 40% of this data was randomly selected and used to train the system at a cutoff 0.05% the data was trained for 34 epochs while 30% of the dataset was used for testing and 12% was used for checking. The stimulation result is as follows.

Table 1: Membership Function, Training Error and Test Error Representation

S/N	Membership Function	Training Error	Test Error
1.	Bell-curve Membership Function	2.47367e-005	0.0002511
2.	Triangular Membership Function	0.0082133	0.0075454
3.	Trapezoidal Membership Function	0.0097643	0.0087654
4.	Guass1 Membership Function	0.0054332	0.005143
5.	Guass2 Membership Function	0.0074532	0.005432

The results were gotten using linear membership function an at an error tolerance level of 0.05

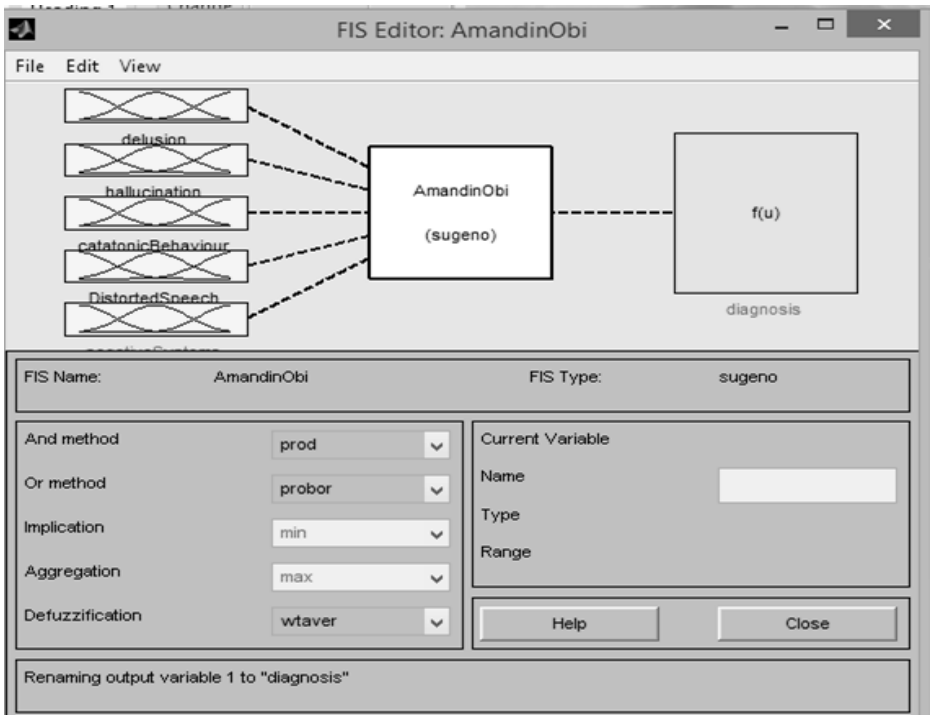


Figure 2: ANFIS Text View Editor

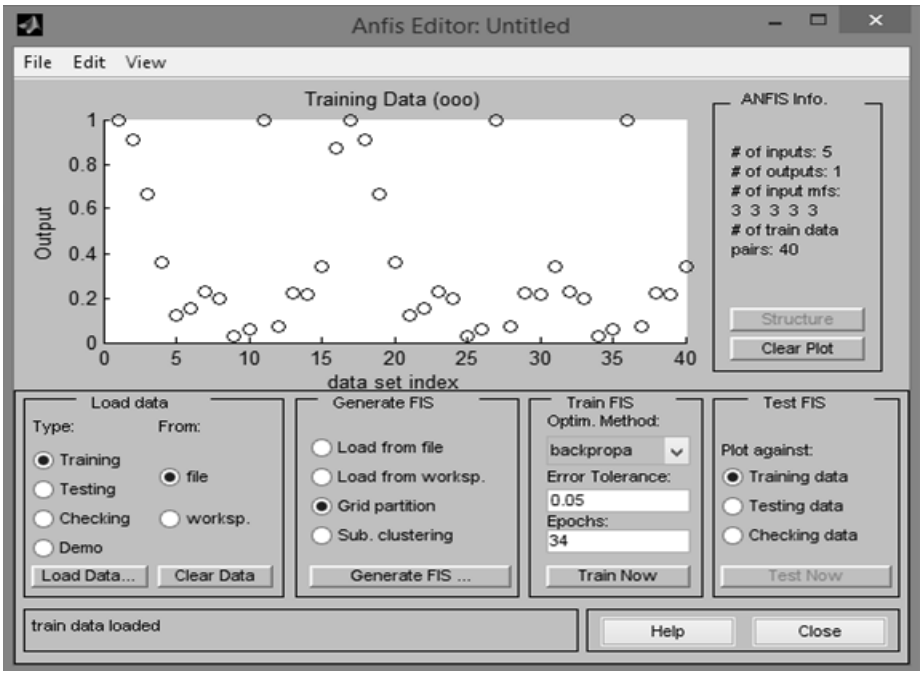


Figure 3: ANFIS Training Data Editor

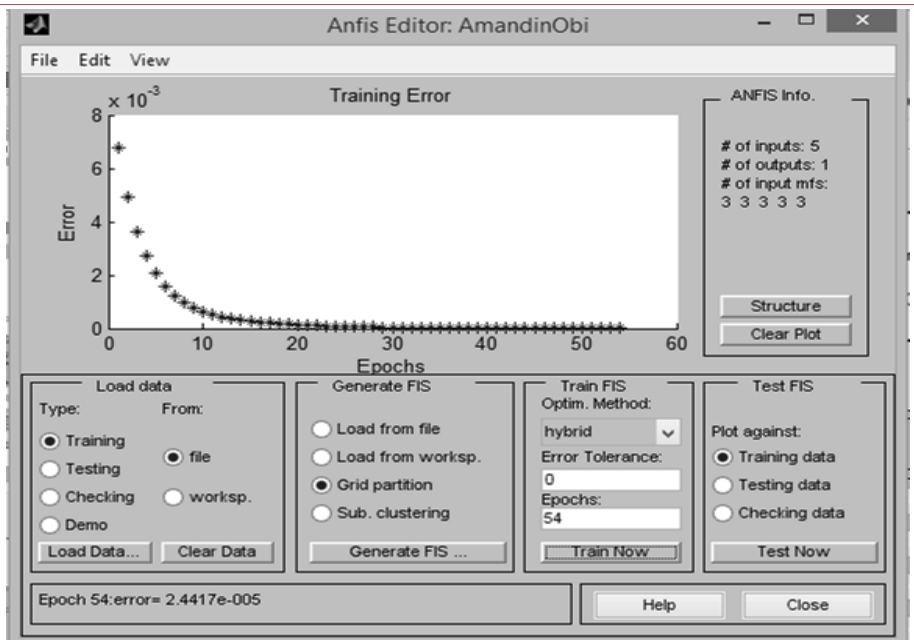


Figure 4: ANFIS Training Error Editor

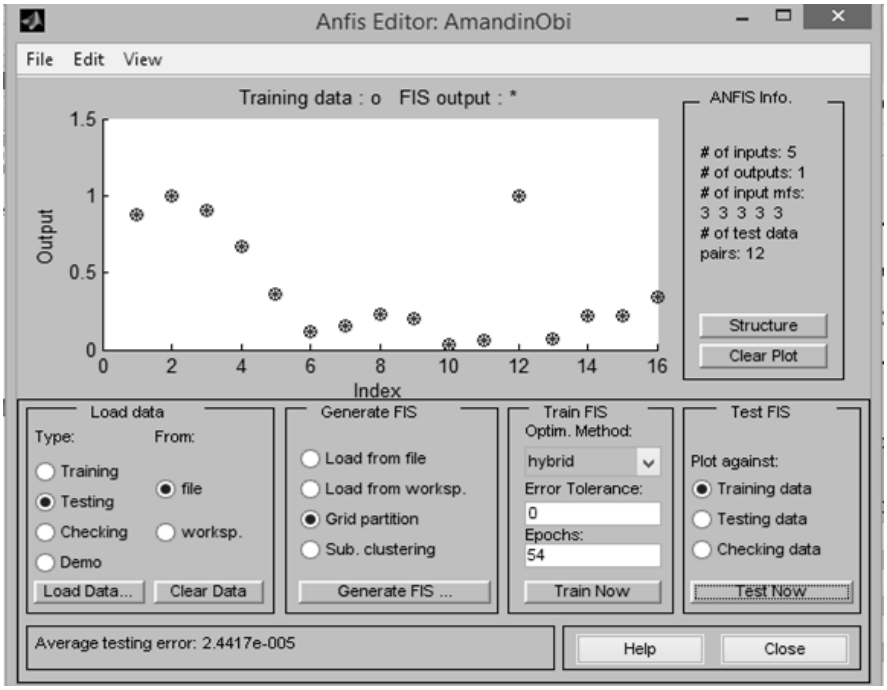


Figure 5: ANFIS Output Editor

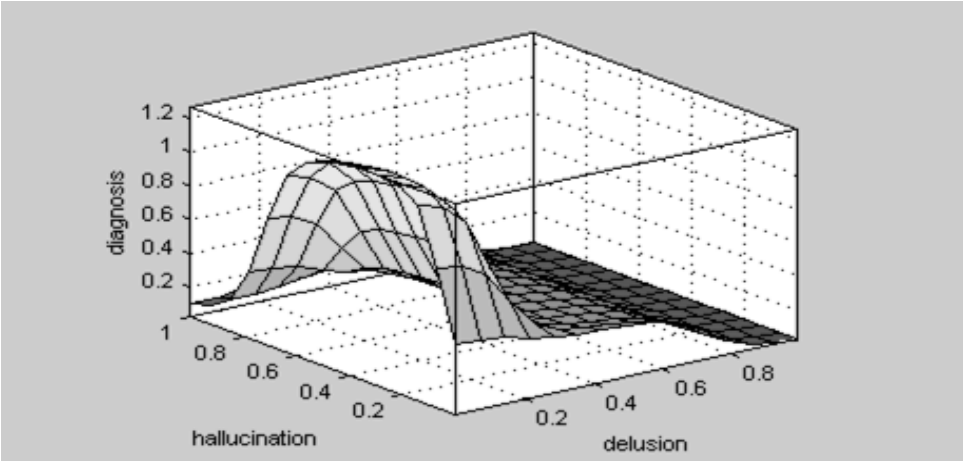


Figure 6: Surface View Illustrating Hallucination and Delusion

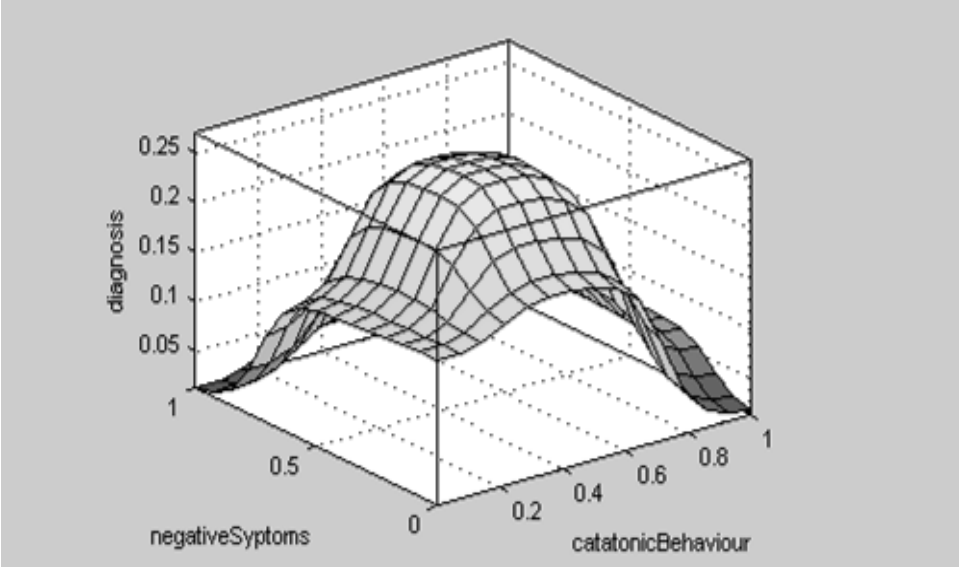


Figure 7: Surface View Illustrating Hallucination and Delusion

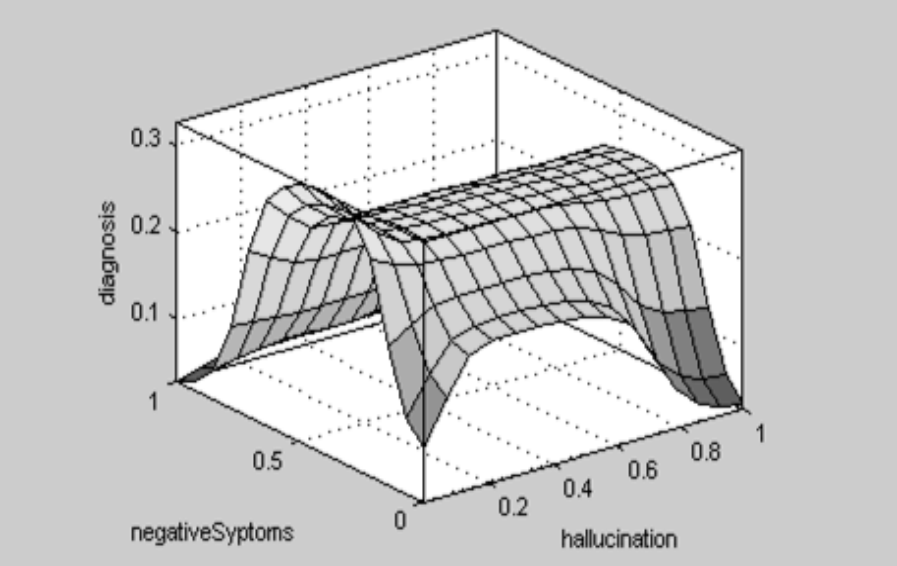


Figure 8: Surface View Illustrating Negative Symptoms and Hallucination

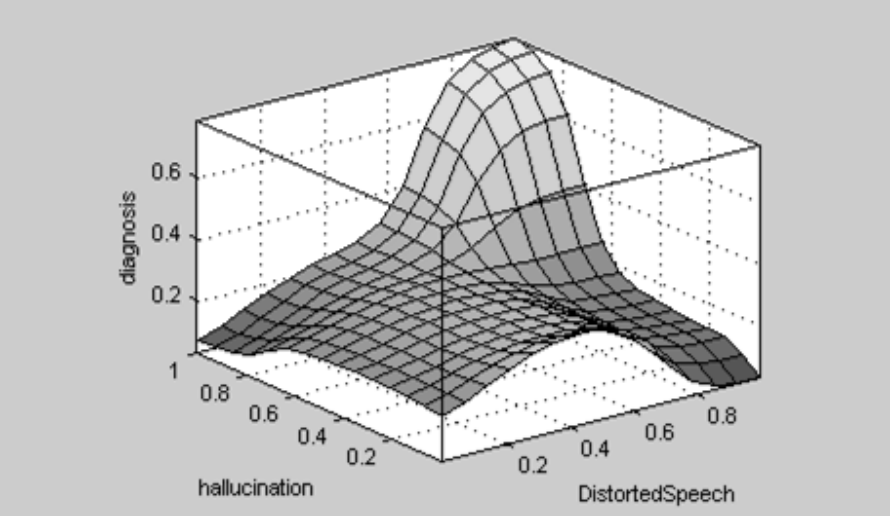


Figure 9: Surface View Illustrating Hallucination and Distorted Speech

5 Conclusion

To draw the inference from the experiment it has been clearly shown that the bell curve membership function shows the least training error when used in training the dataset. The training was completed at 34 epochs and had an error of 2.47367e-005 and the test error which was generated by the ANFIS was 0.0002511. The training was accomplished using a constant membership function type at an error tolerance at 0.05.

REFERENCE

[1] Dwight D., Daniel J. R., Monica E. C., James M. G. and Ruben C. G., (2006), A comparison of cognitive structure in schizophrenia patients and healthy controls using confirmatory factor analysis, Schizophrenia Research 85 (2006) 20 – 29, Available on ScienceDirect.com

[2] Help Guide, (2014), Schizophrenia" retrieved online from HelpGuide.com

[3] Mayo clinic (2014), "Schizophrenia and Treatment" retrieved online from http://www.mayoclinic.org/diseasesconditions/schizophrenia/basics/symptoms/con-20021077

[4] Medicine NET (2014), "Schizophrenia", Retrieved online MedicineNET.Com

[5] NIMH: National Institute for Mental Health (2014), "Schizophrenia" retrieved online www.nimh.nih.gov/health/topics/schizophrenia/

[6] Olivier G. and Marc L. (2005), Neurobiology of Dopamine in Schizophrenia, Departments of Psychiatry and Radiology Columbia University College of Physicians and Surgeons, New York, NY

[7] Right Diagnosis (2014), "Schizophrenia" retrieved online from RightDiagnosis.com

[8] Steffen M. and Todd S. W. (2007), Metacognitive training in schizophrenia: from basic research to knowledge translation and intervention, Curr Opin Psychiatry 20:619–625, Wolters Kluwer Health | Lippincott Williams & Wilkins.

[9] Tara D., Alison C., Janie B. G., Bregje V. S., Katherine F., Kanupriya K. H. Pim C.

[10] and Helen C (2013), Suicide prevention in schizophrenia spectrum disorders and psychosis: a systematic review Donkeret al. BMC Psychology 2013,1:6 retrieved online from http://biomedcentral.com/

[11] Víctor P. and Manuel J.C. (2003), The Diagnosis of Schizophrenia: Old Wine in New Bottles, International Journal of Psychology and Psychological Therapy 2003, Vol. 3, Nº 2, pp. 141-152

[12] WebMD (2014), "Schizophrenia: Diagnosis and Treatment" retrieved online from WebMD.com

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
VOLUME 3, ISSUE 2
ISSN 2054-7390

Evaluation of Tools and Techniques for the Generation of Warning Alerts: A Survey Paper

Abid Ghaffar[1,3], Mohamed Ridza Wahiddin[1], Mohamad Fauzan Noordin[2], Asadullah Shaikh[4]
[1]*Department of Computer Science, Kulliyyah of Information and Communication Technology, International Islamic University, Kuala Lumpur, Malaysia.*
[2]*Department of Computer Science, Kulliyyah of Information and Communication Technology, International Islamic University, Kuala Lumpur, Malaysia.*
[3]*Department of Computer Science, Foundation Year Program, Umm Al-Qura University, Makkah, Kingdom of Saudi Arabia.*
[4]*College of Computer Science and Information Systems, Najran University, King Abdulaziz Road, Najran, Saudi Arabia.*
{mridza, fauzan}@iium.edu.my; aaghaffar@uqu.edu.sa; asshaikh@nu.edu.sa

ABSTRACT

Quality assurance is a key factor for the improvement of an organizational behaviour. It is quite challenging to enhance an organizational performance without realising internal errors and mistakes done by its employees. We have also experienced that most of the security solutions are unsuccessful wherever human behaviour is involved. Organisations sometimes pay huge cost for its survival especially when human error is untraceable and misleading. Online survey has been conducted from different professionals serving at different positions in different organisations. Variety of multi-agent system tools (MAS) is available in the market for modelling and simulation of human behaviour. Brahms modelling and simulation tool has been selected among different multi-agent system tools due to its distinguished features to detect human errors in an organisation which supports warning alert generation system.

Key Words: Brahms Model, Human behaviour Modelling, Cognitive Science, Security and Privacy, Warning Dialogues, Mental Model Approach

1 Introduction

The world is changing and improving day by day with the help of latest developments and research. The real challenge for today's world is to keep in pace of progress and improvement in all the sectors of life. Scientists, technologists, economists, doctors, engineers and professionals are trying their level best to enhance and maintain the quality standards in their existing scenarios. Different sectors of life including agriculture, economy, science, engineering, literature and arts, social sciences need improvement and skills. Every sector of life is involved with human intelligence and behaviour. Human beings are divided into communities and groups based on their religious, social and ethnic beliefs. History proves itself; every impact of society is based on human behavioural changes [17].

Competition in every field of life is a foundation stone for quality assurance and improvement. Every organisation in the existing world wants to improve and produce quality results which are an essential part of its survival. Government and private organisations spend millions of dollars for their own

DOI: 10.14738/tmlai.32.1078
Publication Date: 08th April, 2015
URL: http://dx.doi.org/10.14738/tmlai.32.1078

improvement and enhancement. Sometimes, huge budget is allocated for the machinery and equipment but results are not encouraging and progressive. The human behaviour plays a key-role for the success and improvement of an organisation [22], [30], [31], [32].

This paper addresses two types of surveys, first survey is conducted among those organisations which have human behavioural problems which results in the form of inefficiency and failure of an organisation, second survey is conducted for the selection of tool based on different parameters to detect human behavioural problems in an organisation.

There are different MAS tools available for modelling and simulation of human behaviour but most of the tools do not cover all the aspects of human behaviour especially when human-machine interaction is concerned. Brahms Model provides holistic approach while dealing with human behavioural roles in an organisation keeping in view of human-machine interaction system [26], [27].

Brahms model provides us an opportunity to address the human behaviour problems in an organisation with the help of monitoring human behaviour activities in an organisation [26], [27]. The model is subdivided into sub-models called agents, objects, artifacts, activity, timing, geography, communication and knowledge. It is used to monitor human behaviour activities and warning codes would be used to generate warning alerts by using warning alert generation system. Once human behavioural errors in an organisation are detected well in time, then corrective measures can be taken to improve the system. Consequently, quality assurance in an organisation can be achieved and maximized by using Brahms model along with warning alert generation system [7], [12], [31], [32].

1.1 Contributions

Human behaviour is a key-issue whenever progress of an institution is concerned. Online survey is conducted among different organisations to detect human behaviour problems, which proves that quality assurance problems exist due to human behaviour. Organizational performance is always a key-issue whenever progress of a country is concerned. Online survey proves that different organisations have quality assurance problems due to human behaviour. Organizational performance could be improved and monitored by selecting and using Brahms Modelling and Simulation tool due to its distinguished features which are not available in other MAS [26], [27]. Second survey is conducted for the selection of tool which could detect human behavioural problems in an organisation. Brahms Modelling and Simulation tool is unique due to its completeness in terms of holistic approach. It is found that Brahms Modelling and Simulation tool is ideal to capture human behaviour in any work practice environment. It covers modelling and simulation of human behaviour keeping in view of human-machine interaction system in an organisation [27]. Comparison between Brahms tools with other multi-agent system tools is shown in table 1, which distinguishes the Brahms tool with other MAS tools.

Most of the MAS tools cover one aspect or few aspects of human behaviour in the context of human-machine interaction. Different parameters like Java support, BDI (Belief, Desire, Intention), goal based, Imperative Programming, Subsumption Languages, Holistic Approach, FIPA (Foundation for Intelligent Physical Agents), Declarative and Reactive features are considered to compare Brahms tool with different MAS like Jason, Agent-Speak, Jade, Jadex, Jack and others. The remainder of the paper is structured as follows: Section II presents a survey questionnaire with results. Section III focuses on the factors involved in choosing a language for agent based modelling and Section IV explores the detail

evaluation of existing Brahms modelling and simulations tools. Section V, describes related work. Section VI provides conclusions and identifies directions for future work.

Table 1: Y=Yes, N=No

MAS	Java Support	BDI Based	Goal Based	Imperative	Subsumption Based	Holistic Approach	FIPA Standard	Declarative	Reactive
Brahms	Y	Y	N	N	Y	Y	Y	Y	Y
Jason	Y	Y	Y	Y	N	N	Y	Y	Y
Agent Speak	Y	Y	Y	Y	N	N	Y	Y	Y
Jade	Y	Y	Y	Y	N	N	Y	Y	Y
Jack	Y	Y	Y	Y	N	N	Y	N	Y
Jadex	Y	Y	Y	Y	N	N	Y	Y	Y
Swarm	Y	N	N	Y	N	N	N	N	Y
Repast	Y	N	N	Y	N	N	N	Y	Y
NetLogo	Y	N	N	N	N	N	Y	Y	Y
3APL	Y	Y	Y	Y	N	N	Y	Y	Y
Prolog	N	N	N	N	N	N	N	Y	Y

2 Survey Report

A survey has been conducted to identify the problems in an organisation due to human behaviour. The idea is to determine whether the problems exist in different organisations due to human behavioural activities or not. The questions related to human behaviour are included in the survey and designed in such a manner so that human behaviour problems could be detected in an organisation.

A survey is conducted carrying twenty two questions to address the human behaviour issues in different organisations for various positions. More than hundred respondents have participated in the survey through online survey form. An online survey form was distributed among different countries like Saudi Arabia, United Arab Emirates, Qatar, Pakistan, Malaysia, Germany, Norway, United Kingdom and United States of America. Professionals including directors, managers, engineers and many others working at different positions participated in the survey through online submission form system. We have concluded the following points:

- 75.5% people work in the organisations for their family and 30% people work for money.
- 88% people want to meet the organizational targets and goals.
- 85% people are satisfied from their achieved goals and targets while 9% people are not satisfied from their job objectives.
- 43.5% people use social media up to 2 hours daily.
- 80% people try to follow the job rules while 14% do not follow the job rules.
- 29% people suggested that up to 10% of job rules are not compatible with their job activities while 22% people suggested that up to 20% of job rules are not compatible with their job activities.
- 87% people responded that they do not have health problems while 7% people replied that they have health problems.

- 85% people responded that they discuss issues with their colleagues to resolve problems while 8% people do not like to discuss their problems with their colleagues.
- 91.5% responded in the way that colleagues cooperate with them during job activities while 3% responded negatively in the same sense.
- 77.7% replied that they are well trained and skilled based on the job requirements while 17.5% responded negatively.
- 77.5% people suggested they have all the resources while performing job activities while 17.5% said they do not have enough resources to perform their jobs effectively.
- 38% agreed to work for overtime with less salary offer while 57.5% did not agree to work for overtime without less salary offer.

As a conclusion, the whole survey reveals the larger picture of human behavioural problems which exist in different organisations at different positions and consequently, organizational objectives are compromised. Organisations suffer due to incompatible rules and do not progress due to unskilled and untrained staff. Some people do not interact properly with the colleagues and unable to communicate messages with their colleagues.

3 Factors involved in Choosing A Programming Language for Agent- Based Modelling

Agent-based modelling depends upon the specific requirements. General purpose programming languages, specially designed software and toolkits are used to model agents based on the requirements. Agent Based Modelling System (ABMS) can be developed from smaller scale to the larger scale. Desktop computing for ABMS application development includes Spreadsheets using VBA, dedicated agent-based prototyping environments like Repast, NetLogo and StarLogo [19]. General computational mathematical systems include MATLAB and Mathematica [19]. Large scale agent development environments include Repast, Swarm, Mason, AnyLogic and others. General programming languages used for ABMS include Python, Java and C++ [19].

There are different factors involved in the selection of ABMS. We select ABMS based on certain reasons including natural representation of problem, well defined behaviours and decisions, reflection of agent's behaviour from the actual behaviour of individuals, adaption of agents and change of behaviours, agents learning and dynamic interactions and dynamic relationships of agents [19]. It also includes the modelling of process through which agents form organization, having spatial component to their behaviours and interactions and observation of future without involving the past [19].The number of agents, its interactions and states are also considered. Finally, internal structural changes in the model play a vital role for the selection of ABMS [19].

4 Evaluation of Existing Brahms Modelling and Simulation Tool

We have considered different agent-based modelling and simulation tools which are available in the market to compare with Brahms modelling and simulation aspects. These tools are AgentSpeak, Jason, Jade, Jack, Jadex, JAM, Swarm, Repast, NetLogo, 3APL, Prolog, Soar and ACT-R. These tools are widely used for modelling and simulation of human behaviour as shown in table 1. Brahms stands for "Business Redesign Agent-based Holistic Modelling System", and now it is being used as modelling language which

is comprised of different tools [26], [27]. Brahms provides us the deep insight about human-machine interaction system which helps us to understand how human actually interact with colleagues, documents, and machine, communicate and behave, while performing various job activities [26], [27]. Brahms Process Model comprise of various independent related models which provides us an opportunity to perform modelling job in an easier and efficient way. Brahms uses different models for example, Agents, Objects, Activity, Geography, Timing, Knowledge and Communication which capture all the human-based activities in an organisation [26], [7].

Brahms is considered as an organizational process modelling and simulation tool. Most of multiagent based languages ignore the artifacts and its interaction with environment which makes difficult to develop a holistic model of real-world situations. Brahms actually defines the approach how people and machine work together to accomplish the job by considering behaviours of individuals and groups, how and where communication occurs and synchronization happens in an organisation [26], [27]. Brahms is based on BDI (Belief-Desire-Intention) language and rule-based-system concept [25], [27]. It uses a compiler, a virtual machine to execute Brahms Model, an Eclipse plug-in and an agent viewer program to monitor the activities of different agents at specific interval [23], [27].

According to Brahms modelling and simulation concept, we can record the human behavioural activities in different work practice systems [24]. Brahms is applied in the NASA International Space Stations Mission Control Center (ISSMCC) which is called OCAMS and it is in production since July 2008 [25]. This Program is quite successful despite certain drawbacks which are beyond the control like human mood and emotions.

4.1 AgentSpeak and Jason

It is simple but powerful programming language for building rational agents which is based on BDI paradigm. An intellectual heritage of AgentSpeak involves procedural reasoning systems (PRS) which was developed in late 1980's at Stanford Research Institute (SRI) and logic programming (Prolog).

An implementation of AgentSpeak is called Jason which is a development environment for AgentSpeak. Jason is implemented in Java which includes libraries and debugging tools. The main components of AgentSpeak architecture include Beliefs, Intentions, Desires, Plans and Interpreter. Events are initiated towards Interpreter and then Actions are generated by the Interpreter based on the BDI system [6].

Jason is an extension of AgentSpeak language which is goal based and possesses BDI architecture. The main feature includes its support for persistent belief bases, relatively straight forward distribution over a network, high level speech-act based communications layer and plan labels or annotations which can be used to elaborate functions [6], [9].

Referring to the work practice system, Jason is under gone through experimentation phase for the social simulation. It does not cover all the aspects of simulating a complete work practice system specially when agents are interacting with different objects to perform certain activities at certain location and interval [6].

4.2 Jade (Java Agent Development Framework)

Jade is fully developed in Java and it is used to build agent systems for the management of the networked information resources. It provides a middleware for the development of agent-based applications and can be used in wired and wireless networks. Its main features include interoperability, uniformity, portability, easy to use, and applies the concept of pay-as-you-go-philosophy [8].

Technically, Jade is distributed and multi-party application with peer to peer communication. Jade is object-oriented language and fully compliance with FIPA standards. A single code is written in Jade for database server, application server and presentation client which will work independently and ap¬plied from end-to-end. Different applications of Jade include collaborative work support, e-learning, e-terrorism, network management, knowledge management and entertainment [8].

Simulation of work practice system in an organization like social human interaction, activities like timing, geography, knowledge, communication and objects altogether is not available in Jade [8].

4.3 Jack

It is a commercial and mature product which extends Java in two ways i.e. Syntax and Semantics. Jack defines the top level entities in the form of agent, belief-set, view, event, plan and capability. Applications of Jack include autonomous systems, modelling human-like decision making, decision support applications for military purposes and weather forecasting system [2].

All the features of work practice system which involves human behavioural activities using objects at different situations, locations and timing is not an integral part of Jack [2].

4.4 Jadex

It is a combination of XML and Java using the concept of BDI (Belief-Desire-Intention). Jadex agent consist of two components i.e. ADF (Agent Definition File) and procedural plan code. ADF is written in XML which is based on BDI and procedural plan code is written in Java. Applications of Jadex include teaching and research, portable PDA-based applications, simulation and scheduling [1].

It does not cover the aspects of human dealing within an organization using different objects like computers, devices and documents which are useful to perform certain tasks in the work practice system [1].

4.5 Jam

It is a Java based language comprised of five primary components i.e. world model which represents database, plan library which represent plans to achieve goals by agents, an interpreter which represents the action plan of agent, an intention structure which represents different activities of agents and an observer which is responsible to access plan in order to achieve desired goals [1], [13].

Jam interpreter selects one plan from a list of applicable plans and places it into the intention structure. An agent may or may not execute the newly selected plan based on the existing intention structure and requirements to execute the plan [13].

Jam does not provide us complete picture of work place practice system. Human-centered activities to complete different tasks using objects and artifacts, is not the part of JAM [13].

4.6 Swarm

Swarm intelligence (SI) is an emerging field of biologically-inspired artificial intelligence based on the behavioral models of social insects such as ants, bees, wasps and termites (Bonabeau, 1999).

The main applications include complex interactive virtual environments generation in movie industries, cargo arrangement in airline companies, route scheduling in delivery companies, routing packets in telecommunication networks, power grid optimization control, data clustering, data routing in sensor network, unmanned vehicles controlling in the U.S. military, planetary mapping and micro-satellite controlling in NASA [3], [15], [21].

SI is based on two basic principles which consist of self-organization and stigmergy. There are similarities between distributed computing system and social insects. Biologically inspired computing requires identification of analogies, computer modelling of biological mechanisms, and adaptation of biological mechanisms for IT application including motivation and methods [3], [15], [21].

Swarm is also lacking in human centered approach dealing with different objects and artifacts to perform different activities in an organization [3], [15], [21].

4.7 Repast

Repast stands for Recursive Porous Agent Simulation Toolkit (Repast) which is a free open source toolkit mainly developed by Sallach, Collier, Howe, North and others. It is created in the University of Chicago and it was managed by Argonne National Laboratory. Currently Repast is managed by Repast Organization for Architecture and Development (ROAD). Members from government, industrial sector and academic fields participate in the maintenance of Repast [20].

It borrows many ideas from Swarm agent-based modelling toolkit. It has built in adaptive features like genetic algorithms and regression. Repast can be implemented in different computer languages like Python (Repast Py), Java (Repast J), Dot Net (Repast.Net) and C# which is not possible in Swarm. The main features of Repast include agent templates, fully object-oriented, discrete event scheduling and built-in simulation tools like logging and graphing. It is possible to change the agent behavior equation and agent properties at run time. In addition, Repast provides the support to model the social networking, geographical information system (GIS) and it is available in all modern platforms like Windows, Mac OS and Linux [20].

Modelling and simulation of complete work practice system including agent behaviour and dealing with multiple objects to perform different activities having various parameter is not feasible in Repast [20].

4.8 NetLogo

NetLogo is developed by Northwestern University (Center for Connected Learning and Computer-Based Modelling) which is very easy to setup and run models. A very complicated models are outside the capability of NetLogo but successfully used for abstract models. Once models are created, it becomes harder to extend the model. There is encouraging support from academic community and active maintenance is provided by the software developers. Additional features include 3D visualization of models which can be easily embedded in the web pages [28], [29].

It is a standalone application written in Java which provides multi-agent programming and modelling environment. It is a member of LISP family and supports agents including library of models with variety of domains [28], [29].

NetLogo world comprised of different goals including four types of agents namely Turtles, Patches, Links and Observer. Turtles are agents that move around two dimensional world which is divided into the form of Patches. Directed or undirected Links are the agents which connect two Turtles. Finally, Observer is an agent which does not possess any location [28], [29].

Complete organizational behavior keeping human as center for all the activities using different objects and artifacts is not possible to model and simulate in NetLogo [28], [29].

4.9 3APL

3APL is an agent-based programming with a variant of modal logic. It is a combination of imperative programming and logic programming [4], [5], [11]. The basic components of 3APL are Goals and Beliefs. Applications of 3APL include high level control of mobile robots, small device mobile applications and control behaviour of SONY AIBO robots.

3APL does not cover the capability of modelling and simulating human behaviour in an organisation which perform different tasks using different tools like computers, fax-machines, telephones and specific documents [11].

4.10 Prolog

It is simply defined as programming logic which is most widely used and inspired by logic. The main features of Prolog are writing facts, querying and writing rules [10], [18]. Prolog can be used as query language for relational database and it can be converted into faster and efficient codes. Declaration of variable names is not required; moreover rules are simple and uniform [18].

On the other hand, prolog is slower and provides an unnatural way to program which can be understood by only expert programmers. It seems clumsy for numerical calculations which cannot be implemented in fast hardware. Prolog cannot be implemented in faster hardware and consumes lot of memory. It also lacks in real time capability.

The main applications of prolog include race track applications, compiler design, digital electronic circuit verification and providing program correctness. Prolog does not provide us complete picture of organisational behaviour in the form of modelling and simulation, where humans are involved in certain activities using different objects and artifacts [10].

4.11 Soar

It provides general cognitive architecture for developing systems which possess intelligent behaviour. It carries full range of tasks which can represent and use knowledge properly. Problem solving methods are employed successfully. Different aspects of tasks are learned and its performance is checked and monitored. The combination of different relevant knowledge reveals the decisions at run time process. All decisions made are based on interpretation of sensory data, context of working memory and relevant knowledge. Decisions are not interrupted into the form of sequences [16].

Work practice system in an organisation cannot be modeled and simulated in Soar, where human social behaviour is involved specially performing series of different tasks to achieve targeted goals using different objects is a question mark [16].

4.12 ACT-R

It is simply defined as human cognitive architecture, how human thinking-process works actually, and generates knowledge. ACT-R is simulation of understanding human cognition. There is a comparison between ACT-R cognitive modal and fMRI (functional magnetic resonance imaging) dataset which is used to predict the results based on datasets of fMRI [14].

Complete human behavioural modelling and simulation, when human is performing different activities in an organisation using different tools and objects is not available in ACT-R [14].

Most of the tools available in the market for modelling and simulation of human behaviour are limited only to agent-based modelling and simulation as shown in table 1, but Brahms provides us an opportunity not only to model but also simulate complete work practice system in an organisation where human role is significant [24].

5 Related Work

Charles M. Macal et al. (2009) presented an idea about agent based modelling system, its application and development process in different aspects of practical scenarios. He focused on ABMS usefulness and its advantage over conventional systems. Applications of ABMS includes Air Traffic Control System, Crime Analysis, Biomedical Research, Chemistry, Epidemic Modelling, Organisational Decision Making and Market Analysis. Different toolkits used for ABMS are Netlogo, Starlogo, Swarm, Repast Simphony, Matlab and Mathematica [19].

Maarten Sierhuis et al. (2002) presented an idea about modelling and simulating work practice system in an organisation using Brahms Model. Brahms Model is comprised of further sub-models called Agents, Objects, Activity, Geography, Timing, Knowledge and Communication. All the activities in a work practice system could be monitored at any interval using Agent Viewer [26].

Chin Seah et al. (2005) gave an idea about the analysis of Brahms model tool by its application in the NASA Mars Exploration Rover (MER) mission. There is a complex situation whenever there is an analysis of work practice system keeping in view of human-machine interaction system. It's quite challenging to model and simulate space mission program using Brahms Model as still there are some areas where we need to improve Brahms modelling and simulating tool [23].

Maarten Sierhuis et al. (2007) presented an idea about detailed version of Brahms Modelling and Simulation technique at the implementation level where more details are needed observe people behaviour and activities in an organisation. It was proved that Brahms Modelling and Simulation technique could be applied in large organisations [27].

Maarten Sierhuis (2013) presented an idea about modelling and simulating multi-agents using Brahms Model which involve people behaviour and their activities at different levels. In contrast, SOAR and ACT-R focused on individual agent and its behaviour. Example of one day work practice system was considered to establish the facts about Brahms Modelling and Simulation [25].

F. Bellifemine et al. (2003) presented detailed information about the Jade Platform. Main functional-ities of Jade, its architecture and its conceptual model were discussed. Two basic components of its conceptual model comprised of Jade distributed system topology and software component architecture with agent paradigm were explained. Network topology defines, how different components linked together? [8].

Rafael H. Bordini et al. (2006) presented an idea about programming languages and development tools for multi-agent systems. He discussed programming languages issues in terms of declarative, imperative and hybrid nature. He also focused on integrated development environment, platform and framework for these languages [4].

Bijaya Ketan Panigrahi,Yuhui Shi et al. (2011) presented detailed information about biologically inspired algorithms called swarm intelligence. It contains algorithms based on collective individual behaviours and finding its optimal solution. The book covers existing research in its own kind and its real world applications [21].

Mehdi Dastani et al. (2012) presented proceedings of the International Workshop on Programming Multi- Agent Systems (ProMAS 2012) where programming languages and tools for MAS were discussed. Theory and practical issues were discussed in detail in order to analyse different techniques and concepts of multi-agent systems. Revised published papers were presented as source of information [9].

Rafael H. Bordini et al. (2007) presented detailed information about AgentSpeak language using Jason. The book contains explanation about programming multi-agent system using programming language calledAgentSpeak with the help of Jason [6].

Agent Oriented Software Limited Group called AOS group presented white paper carrying detailed information about Jack functionalities and its applications. AOS group focused the application areas where autonomous decision making process is required like Oil Production Systems and Unmanned Vehicles on exploration missions both for under water and space. Jack is autonomous, efficient, resilient and carrying rapid specification [2].

Rafael H. Bordini et al. (2009) presented published papers in the form of book, “Multi-Agent Programming: Languages, Tools and Applications”, which comprised of published papers information about multi-agent systems [1].

Alexander Pokahr et al. (2005) presented detailed information about Jadex BDI (Belief, Desire, Intention) reasoning engine. Jadex is a combination of XML (Extensible Markup Language) and Java using BDI concepts. Jadex platform supports editing, debugging and execution of multi-agent systems [1].

Marcus J. Huber et al. (1999) presented idea about JAM architecture and its mobile operational capability due to Java implementation. JAM is a hybrid architecture based on Procedural Reasoning System (PRS), Structured Circuit Semantics (SCS) and Plan-Action combination [13].

Christian Blum et al. (2008) expressed his views about swarm intelligence starting from basic concepts up to the level of real world applications. The book covers foundation concepts of swarm intelligence and presents applications in the field of swarm robotics and telecommunication technology [3].

Aleksandar Jevti’ (2011) presented his doctoral thesis about swarm intelligence general framework. He proposed general new design methodology for swarm intelligence tools which did not exist before [15].

Michael J. North et al. (2006) expressed his views about standardization of simulation architecture keeping in view the varying standards followed by different authors. Repast toolkit was examined in different context and suggestions were made for its improvement [20].

Seth Tisue et al. (2004) presented his views about multi-agent language called NetLogo which is being used for education and research. NetLogo is a multi-agent programming language which provides modelling environment for the simulation of complicated phenomenon [28].

Uri Wilensky (1999) presented user manual for NetLogo version 5.0.5 which contains detailed information at the user level. It is in continuous process of development at Center for Connected Learning and Computer-Based Modelling. Instructions can be given to thousands of agents operating at different levels [29].

Fernando Koch et al. (2005) presented his idea about 3APL-M platform which is used to provide multi-agent system development environment on mobile devices which has limited resources. BDI and Java language was used in the development of 3APL-M [11].

Ivan Bratko et al. (1986) presented his ideas about programming language which is termed as Prolog. It is non-procedural and declarative language which makes the job easier for the programmer to perform logic operations. The main features include pattern matching, tree-based data structuring and automatic backtracking [18].

Fernando C. N. Pereira et al. (2002) presented his ideas about computational linguistics and logic programming. It focuses on the main concepts of definite clause formalism used in computational linguistics and logic programming [10].

John E. Liard (2012) expressed his views about design and structure of SOAR which is based on complex cognitive architecture. Detailed study has been done, how SOAR works actually and how it respond in a given environment [16].

Jelmer P. Borst et al. (2014) presented his idea about comparison of brain thinking process using ACT-R and FMRI (Functional Magnetic Resonance Imaging). The focus remains in the use of ACT-R architecture using FMRI data [14].

Stephen P. Robbins et al. (2012) expressed his views about the impact of human behaviour in an organisational behaviour. He focused on different aspects of human behaviour in order to enhance the organisational performance [22].

John Preece (2013) presented his idea about human behaviour and its background. He focused on human behavioural roles and its impact on human society [17].

Abid Ghaffar et. al (2013) presented his idea about warning alert generation system using Brahms Mod¬elling and Simulation technique. Human behaviour in an organisation could be improved and monitored and rapid action could be taken in the right direction [12].

6 Conclusion and Future Work

Organisational performance is compromised due to human behaviour and challenging to address human behavioural roles in terms of performance and improvement. Online survey has been conducted which

clearly shows human behavioural problems in different organisations. Different MAS tools are evaluated for modelling and simulation of human behaviour in an organisation and we find Brahms modelling and simulation tool is the most appropriate and compatible for the simulation of Work practice system in an organisation. We can generate warning alerts using Brahms Modelling and Simulation tool based on its unique features which is not available in other MAS. MAS have played a significant role in the development of every field of life. Information technology would not be considered consistent, if MAS is not used and applied correctly in the progress of an industry. There is a need to check all the features of MAS in more details as MAS are being developed and updated on regular basis. We can incorporate those changes and developments in our warning alert generation system.

ACKNOWLEDGEMENT

This research is partially funded by the Malaysian Ministry of Education grant ERGS 11-010-0010 and partially funded by the Umm Al-Qura University, Makkah, Kingdom of Saudi Arabia. We would like to thank Dr. Ghassan Nauman for his useful support.

REFERENCES

[1] El Fallah Seghrouchni, A., Dix, J., Dastani, M., & Bordini, R. H. (2009). Multi-Agent Programming. Multi-Agent Programming:: Languages, Tools and Applications, ISBN 978-0-387-89298-6. Springer-Verlag US, 2009, 1.

[2] AOS Limited. An Agent Infrastructure for Providing the Decision-Making Capability Required for Autonomous Systems, 2013.

[3] Blum, C., & Li, X. (2008). Swarm intelligence in optimization (pp. 43-85). Springer Berlin Heidelberg.

[4] Bordini, R. H., Braubach, L., Dastani, M., El Fallah-Seghrouchni, A., Gomez-Sanz, J. J., Leite, J., ... & Ricci, A. (2006). A survey of programming languages and platforms for multi-agent systems. Informatica (Slovenia), 30(1), 33-44.

[5] Dix, J., & Seghrouchni, A. E. F. (2005). Multi-Agent Programming. R. H. Bordini, & M. Dastani (Eds.). Springer Science+ Business Media, Incorporated.

[6] Bordini, R. H., Hübner, J. F., & Wooldridge, M. (2007). Programming multi-agent systems in AgentSpeak using Jason (Vol. 8). John Wiley & Sons.

[7] Bravo-Lillo, C., Cranor, L. F., Downs, J. S., & Komanduri, S. (2011). Bridging the gap in computer security warnings: A mental model approach. IEEE Security & Privacy, 9(2), 0018-26.

[8] Caire, F. B. G., Poggi, A., & Rimassa, G. (2003). JADE. A white paper.

[9] Dastani, M., Hbner, J. F., & Logan, B. (2013). Programming Multi-Agent Systems: 10th International Workshop, ProMAS 2012, Valencia, Spain, June 5, 2012, Revised Selected Papers. Springer Publishing Company, Incorporated.

[10] Pereira, F. C., & Shieber, S. M. (2002). Prolog and natural-language analysis. Microtome Publishing.

[11] Fernando Koch. 3APLM Platform for Deliberative Agents in Mobile Devices, 2005.

[12] Ghaffar, A., Wahiddin, M. R., & Shaikh, A. (2013). Computer Assisted Alerts Using Mental Model Approach for Customer Service Improvement. Journal of Software Engineering and Applications, 6(05), 21.

[13] Huber, M. J. (1999, April). JAM: A BDI-theoretic mobile agent architecture. In Proceedings of the third annual conference on Autonomous Agents (pp. 236-243). ACM.

[14] Borst, J. P., & Anderson, J. R. (2014). Using the ACT-R Cognitive Architecture in combination with fMRI data. An Introduction to Model-Based Cognitive Neuroscience. Springer, New York.

[15] Jevtić, A. (2011). Swarm intelligence: novel tools for optimization, feature extraction, and multi-agent system modeling (Doctoral dissertation, Telecomunicacion).

[16] Laird, J. (2012). The Soar cognitive architecture. MIT Press.

[17] J. Preece. A Brief History of Human Behaviour and How to Become an Enlightened Global Citizen (Smashwords Edition, 2013).

[18] Bratko. Prolog Programming For Artificial Intelligence, Addison-Wesley, 1986.

[19] Macal, C. M., & North, M. J. (2009, December). Agent-based modeling and simulation. In Winter simulation conference (pp. 86-98). Winter Simulation Conference.

[20] North, M. J., Collier, N. T., & Vos, J. R. (2006). Experiences creating three implementations of the repast agent modeling toolkit. ACM Transactions on Modeling and Computer Simulation (TOMACS), 16(1), 1-25.

[21] Panigrahi, B. K., Shi, Y., & Lim, M. H. (2011). Handbook of swarm intelligence: concepts, principles and applications (Vol. 8). Springer Science & Business Media.

[22] Robbins, S., Judge, T. A., Millett, B., & Boyle, M. (2013). Organisational behaviour. Pearson Higher Education AU.

[23] Seah, C., Sierhuis, M., and J. C., Clancey. Multi-agent modeling and simulation approach for design and analysis of MER Mission Operations. In Proceedings of 2005 International

conference on Human-Computer interface advances for modeling and simulation (SIMCHI 2005), pages 73–78. Citeseer, 2005.

[24] Sierhuis, M., Modeling and simulating work practice: BRAHMS: A multiagent modeling and simulation language for work system analysis and design. Ph.D Thesis, UvA-DARE, University of Amsterdam (UvA) 2001.

[25] Sierhuis, M. (2013). Multi-agent activity modeling with the Brahms environment. In Theory, Practice, and Applications of Rules on the Web, pages 34–35. Springer Berlin Heidelberg.

[26] Sierhuis, M., & Clancey, W. J. (2002). Modeling and simulating practices, a work method for work systems design. Intelligent Systems, IEEE, 17(5), 32-41.

[27] Sierhuis, M., Clancey, W. J., & Van Hoof, R. J. (2007). Brahms: a multi-agent modelling environment for simulating work processes and practices. International Journal of Simulation and Process Modelling, 3(3), 134-152.

[28] Tisue, S., & Wilensky, U. (2004, May). Netlogo: A simple environment for modeling complexity. In International conference on complex systems (pp. 16-21).

[29] Uri Wilensky. NetLogo User Manual, version 5.0.5, 2013.

[30] M. F. Noordin. (2013). ICT and Islam, IIUM Press.

[31] Ghaffar, A., Wahiddin, M. R., Noordin, M. F., & Shaikh, A. (2015). A Framework to Improve Customer Service Using Brahms Model. IJEIR, 4(1), 99-106.

[32] Ghaffar, A., Wahiddin, M. R., Shaikh, A., and Ahmad, A. (11-13 Feb. 2015). Generating Alerts using context aware security and Brahms Model for customer service improvement. Accepted paper in International Multi-Topic Conference, Mehran University, Jamshoro, Pakistan. IMTIC'15.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
VOLUME 3, ISSUE 2
ISSN 2054-7390

Frame Based Postprocessor for Speech Recognition Based on Augmented Conditional Random Fields

[1]Yasser Hifny
[1]Faculty of computers and information systems, University of Helwan, Egypt
yhifny@fci.helwan.edu.eg

ABSTRACT

In this paper, we present a novel postprocessor for speech recognition using the Augmented Conditional Random Field (ACRF) framework. In this framework, a primary acoustic model is used to generate state posterior scores per frame. These output scores are fed to the ACRF post processor for further frame based acoustic modeling. Since ACRF explicitly integrates acoustic context modeling, the postprocessor has the ability to discover new context information and to improve the recognition accuracy. The results on the TIMIT phone recognition task show that the proposed postprocessor can lead to significant improvements especially when Hid-den Markov Models (HMMs) were used as primary acoustic model. Keywords: Hidden Markov models; augmented conditional random fields; deep conditional random fields; speech recognition postprocessor.

1 Introduction

Acoustic modeling post processing based on methods derived from Conditional Random Fields [1] is an active area of research [2], [3], [4]. CRFs have a generic way to define feature functions (constraints). Consequently, the feature functions play a vital role in defining the model and its applications [5]. In this work, we present a frame based postprocessor for speech recognition based on ACRFs [6, 7]. The ACRFs paradigm is a nonlinear variant of CRFs where the feature functions are computed from scoring a large number of Gaussians. The projection of low dimensional acoustic data into a high dimensional (augmented) space aims to simplify the classification problem. The main advantage of this framework is that acoustic context information is explicitly integrated to handle the sequential phenomena of the speech signal and hence can be expected to improve the recognition accuracy. The ACRFs can be efficiently estimated using the Approximate Iterative Scaling (AIS) algorithm.

In the original ACRF framework, the process of augmenting the low dimensional space to obtain a high dimensional space (o_t ! o_t^{Aug}) is based on the following algorithm:

1. A large number of Gaussians is estimated from the training data using the EM algorithm [8].
2. The Gaussians provide scores for each frame.

DOI: 10.14738/tmlai.32.943
Publication Date: 14th April, 2015
URL: http://dx.doi.org/10.14738/tmlai.32.943

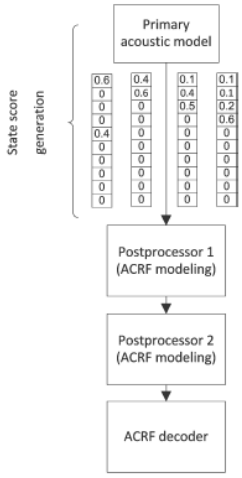


Figure 1: Frame based postprocessor using augmented conditional random field (ACRF) framework.

3. The scores are sorted and only the n-best scores are retained to reduce the storage requirements during the training. Typically, the n-best nearest-neighbor shortlist size is set to 10.
4. An augmented vector is constructed and its size dAug equals the number of Gaussians in the recognition problem. A state feature value is calculated as a pruned posterior score for each Gaussian and is given by

$$b_i(\mathbf{o}_t) = \frac{\mathcal{N}_i(\mathbf{o}_t;\lambda)}{\sum_j \mathcal{N}_j(\mathbf{o}_t;\lambda)} \approx \frac{\mathcal{N}_i(\mathbf{o}_t;\lambda)}{\sum_{j\in n-\text{best}} \mathcal{N}_j(\mathbf{o}_t;\lambda)} \tag{1}$$

Where $N_i(o_t; \lambda) \approx 0$ for $i \notin n-$best list and the normalization step is conceptually redundant to improve the ACRFs training speed. Frame based acoustic models generate state scores. These state scores are fed to a decoder to generate the recognition hypothesis. For example, in HMMs [9, 10, 11, 12], an acoustic feature vector o_t may be generated, with an output probability density function $b_j(o_t)$, which is associated with state j. A mixture of Gaussian distributions is typically used to model the output distribution for each state,

$$b_j(\mathbf{o}_t) = \sum_{m=1}^{M} c_{jm}\mathcal{N}(\mathbf{o}_t;\mu_{jm},\Sigma_{jm}) \tag{2}$$

Where M is the number of mixture components, c_{jm} is the component weight and $\sum_m^M C_{jm} = 1$. μ_{jm} and Σ_{jm} are the component specific mean vector and covariance matrix respectively. These state scores can be sorted and normalized in a similar way as in Equation (1). Hence, the normalized state scores is given by:

$$\mathfrak{N}_i(\mathbf{o}_t) = \frac{\sum^{s} \rho^{s}(\mathbf{o}_t)}{\rho^{i}(\mathbf{o}_t)} \approx \frac{\sum^{k\in n-\text{best}} \rho^{k}(\mathbf{o}_t)}{\rho^{i}(\mathbf{o}_t)} \tag{3}$$

Where $b_j(o_t) \approx 0$ for $j \notin n-$best list.

The generated normalized state scores in Equation (3) are fed to ACRF postprocessor for further acoustic modeling. The ACRF output state scores can be normalized in a similar way and fed to a second layer ACRF for further acoustic modeling. An example of the described process is shown in Figure 1. By explicitly integrating acoustic context modeling using the ACRFs, the post-processors have the ability to discover new context information and to improve the recognition accuracy. This is the main motivation behind the work. In this work, we investigated three different primary acoustic models which have different modeling power[1]. In particular, HMMs were tested as the main acoustic model. In addition, ACRF acoustic modeling as described in [7] was evaluated as a primary acoustic model. Finally, powerful deep conditional random fields (DCRFs) [13] were developed as a primary acoustic model. DCRFs are a variant of hybrid deep neural networks DNN/HMM [14], [15], [16], [17], [18] formulated using the maximum entropy principle [19]. The main goal of testing different primary acoustic models is to show the modeling effect of using an ACRF postprocessor.

This paper is organized as follows: the mathematical formulation of ACRFs is given in Section 2. Section 3 describes how to compute the normalized state scores for different primary acoustic models. Experimental results on a phone recognition task are given in Section 4. Finally, a summary of the presented work is given in the conclusions.

2 Augmented Conditional Random Fields

ACRFs are undirected graphical models that maintain the Markov properties of HMMs. They operate in a high dimensional (augmented) space to improve the discrimination between speech classes. This augmented space is constructed by

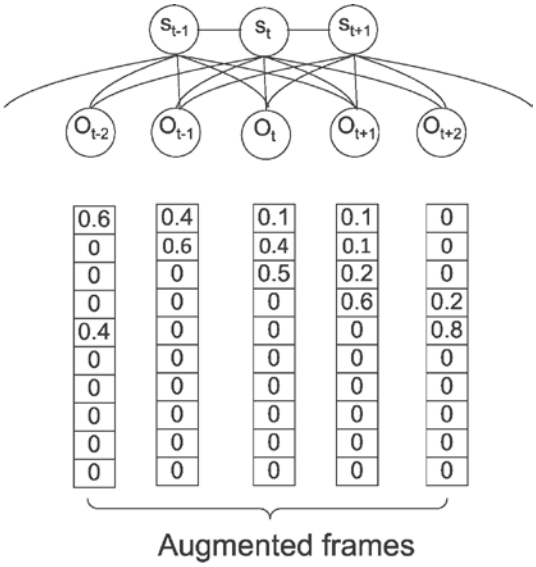


Figure 2: ACRF phone model with state scores computed from a window of augmented frames.

[1] A primary acoustic model provides the features to the postprocessors.

scoring a large number of Gaussians. In addition, by using a large window of augmented frames, acoustic context information is explicitly integrated allowing the model to handle the sequential nature of speech signals. Hence, the HMM conditional independent assumptions are relaxed in this framework. ACRFs

Feature functions are based on pruned posterior scores to improve the training speed. The ACRFs have a batch training algorithm that scales for a large amount of training data. The linear chain undirected graphical model behind the ACRF is shown in Figure 2. The model has the following properties:

- It obeys the Markovian property.
- The state scores are computed from the augmented frames (pruned posterior scores).

Given a state sequence S = (s_1, s_2……s_T) and a time sequence of speech frames or acoustic observations associated an utterance O = (o_1, o_2…..o_T),the maximum entropy conditional distribution de_ning ACRFs is

$$P_\Lambda(\mathbf{S}|\mathbf{O}) = \frac{1}{Z_\Lambda(\mathbf{O})} \prod_{t=1}^{T} \exp\left(\lambda_{s_t s_{t-1}} a(s_t, s_{t-1}) + \sum_{u=t-c}^{t+c} \sum_{i=1}^{d^{\mathrm{Aug}}} \lambda_{s_t}^{ui} x_i(o_u)\right) \quad (4)$$

where $\lambda_{S_t}^{ui}$ and $\lambda_{S_t S_{t-1}}$ are associated with the feature functions $x_i(O_u)$ and the transition functions $a(S_t, S_{t-1})$. The feature functions $x_i(O_t)$ [2]are computed as in Equation (3) when HMMs are used as a primary acoustic model. The number of frames in the acoustic context window is w = 2c+1. $Z_\Lambda(O)$ (Zustandsumme) is a normalization coefficient referred to as the partition functions and is given by

$$Z_\Lambda(\mathbf{O}) = \sum_{\mathbf{S}} \prod_{t=1}^{T} \exp\left(\lambda_{s_t s_{t-1}} a(s_t, s_{t-1}) + \sum_{u=t-c}^{t+c} \sum_{i=1}^{d^{\mathrm{Aug}}} \lambda_{s_t}^{ui} x_i(o_u)\right) \quad (5)$$

and it can be computed efficiently using the forward algorithm [1]. The feature functions $x_i(o_t)$ are computed in a different way for other primary acoustic models. Section 3 will explain how to compute these feature functions for ACRFs and DCRFs acoustic models. In particular, Equation (11) and Equation (16) are used for primary acoustic models based on ACRFs and DCRFs respectively. The primary acoustic decoding results are based on state scores. Compared to the primary system, the ACRF post-processing sees next to the current set of state scores also those of the neighboring frames, allowing the integration of context information in the augmented space. It is worth to mention that when acoustic context information is not modeled (i.e. c = 0), the ACRF post-processor and the primary acoustic model should lead to the same recognition results.

2.1 ACRF Optimization

For *R* training observations {O_1,O_2,O_r, …O_R} with corresponding transcriptions *{W_r},* ACRFs are trained using the conditional maximum likelihood (CML) criterion to maximize the posterior probability of the correct word sequence given the acoustic observations. Exact lower bound optimization algorithms for CRFs are very slow [1]. Therefore, we use the Approximate Iterative Scaling (AIS) algorithm to speed up

[2] $a(S_t, S_{t-1})$. is binary valued and can be used to specify the transition topology.

the training process. The value of the learning rate is the main difference between exact and approximate algorithms. An AIS algorithm update equation is given by:

$$\lambda_{ji}^{\tau+1}(\mathbf{O}) = \lambda_{ji}^{\tau}(\mathbf{O}) + \eta_{\text{AIS}} \log \frac{\mathcal{C}_{ji}^{\text{num}}(\mathbf{O})}{\mathcal{C}_{ji}^{\text{den}}(\mathbf{O})}, \tag{6}$$

where $\eta_{AIS} = \frac{1}{w}$ is called the learning rate and τ is the iteration number. The sparse accumulators of the sufficient statistics, $C_{ji}(O)$, for the J^{th} state and i^{th} constraint are calculated as follows:

$$\mathcal{C}_{ji}^{\text{num}}(\mathbf{O}) = \sum_{r=1}^{n} \sum_{t=1}^{T_r} \gamma_j^r(t|\mathcal{M}^{\text{num}}) \mathbf{o}_{rti}^{\text{Aug}} \tag{7}$$

$$\mathcal{C}_{ji}^{\text{den}}(\mathbf{O}) = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \gamma_j^r(t|\mathcal{M}^{\text{den}}) \mathbf{o}_{rti}^{\text{Aug}}. \tag{8}$$

Where r is the utterance index and $O_t^{Aug} = [O_{t-c}, \ldots O_t, \ldots O_{t+c}]$. Given the forward score $\alpha_j(t)$ and backward score $\beta_j(t)$, the occupation probability of being in state J at time t, γ_j, is given by:

$$\gamma_j(t|\mathcal{M}) = P(\mathbf{s}_t = j|\mathbf{O}; \mathcal{M}) = \frac{\alpha_j(t|\mathcal{M})\beta_j(t|\mathcal{M})}{Z_\Lambda(\mathbf{O}|\mathcal{M})} \tag{9}$$

and to avoid the necessity of building lattices, the $\gamma_j(t \mid M)$ is approximated with state estimates as follows [20]:

$$\gamma_j(t|\mathcal{M}) = \frac{\exp\left(\sum_{u=t-c}^{t+c} \sum_{i=1}^{d^{\text{Aug}}} \lambda_j^{ui} x_i(\mathbf{o}_u)\right)}{\sum_{\mathbf{s}} \exp\left(\sum_{u=t-c}^{t+c} \sum_{i=1}^{d^{\text{Aug}}} \lambda_{\mathbf{s}}^{ui} x_i(\mathbf{o}_u)\right)} \tag{10}$$

3 State scores generation

Three di_erent primary acoustic models which have di_erent modeling power were developed in this work. For HMM, the generated normalized state scores are computed as in Equation (3). For ACRF and DCRFs, the goal of this section is to show how to compute their normalized state scores.

3.1 ACRF as a primary acoustic model

ACRFs can be used as a primary acoustic model if the input features to ACRFs are based on Equation (1) . The parameter estimation is exactly identical to described in Section 2. The normalized state scores are given by

$$x_j(\mathbf{o}_t) = \frac{\exp\left(\sum_{u=t-c}^{t+c} \sum_{i=1}^{d^{\text{Aug}}} \lambda_j^{ui} b_i(\mathbf{o}_u)\right)}{\sum_{\mathbf{s}} \exp\left(\sum_{u=t-c}^{t+c} \sum_{i=1}^{d^{\text{Aug}}} \lambda_{\mathbf{s}}^{ui} b_i(\mathbf{o}_u)\right)} \approx \frac{\exp\left(\sum_{u=t-c}^{t+c} \sum_{i=1}^{d^{\text{Aug}}} \lambda_j^{ui} b_i(\mathbf{o}_u)\right)}{\sum_{k \in n-\text{best}} \exp\left(\sum_{u=t-c}^{t+c} \sum_{i=1}^{d^{\text{Aug}}} \lambda_k^{ui} b_i(\mathbf{o}_u)\right)} \tag{11}$$

where $\exp\left(\sum_{u=t-c}^{t+c} \sum_{i=1}^{d^{\text{Aug}}} \lambda_j^{ui} b_i(\mathbf{o}_u)\right) \approx 0$ for $j \notin n-\text{best list}$.

3.2 DCRF as a primary acoustic model

Training CRFs on the top of a hidden layer constructed from scoring a large number of sigmoid functions was introduced in [17]. One way to improve this approach is to compute the state scores based on a DNN that has several hidden layers [21]. Deep Conditional Random Fields acoustic models are a particular implementation of linear chain CRFs where the state scores are computed based on a DNN that has several hidden layers [13]. The output layer of DCRFs is based on linear activation functions while in hybrid DNN/HMM it is based on softmax activation functions. This is the main difference between DCRFs and hybrid DNN/HMM systems. A graphical representation of the DCRF acoustic model is shown in Figure 3. The conditional distribution defining DCRFs is given by

$$P_\Lambda(\mathbf{S}|\mathbf{O}) = \frac{1}{Z_\Lambda(\mathbf{O})} \prod_{t=1}^{T} \exp\left(\lambda_{\mathbf{s}_t \mathbf{s}_{t-1}} a(\mathbf{s}_t, \mathbf{s}_{t-1}) + b_{\mathbf{s}_t}(\mathbf{o}_t)\right) \tag{12}$$

Where $b_{S_t}(o_t)$ is computed from a DNN scorer.

The feed-forward phase of a DNN scorer updates the output value of each hidden unit. Each hidden unit output is computed as follows:

$$\mathbf{o}_{tj}^{h} = \mathrm{sigm}(\sum_{i=1}^{n} \lambda_{ij} \mathbf{o}_{ti}^{h-1}) \tag{13}$$

Where O_t^h is an output of a hidden layer, n is the number of inputs, and h is an index to a hidden layer. The sigmoid function is computed as follows:

$$\mathrm{sigm}(x) = \frac{1}{1+e^{-x}} \tag{14}$$

The output of a hidden layer is forwarded to the next layer until the output layer is computed as follows (linear activation):

$$\mathbf{o}_{tj}^{N} = \sum_{i=1}^{n} \lambda_{ij} \mathbf{o}_{ti}^{N-1} \tag{15}$$

where N is the index of the output layer. Hence, $b_{S_t}(O_t) = O_{ts_t}^N$ connects a DNN scorer to CRFs.

The normalized state scores are given by

$$x_j(\mathbf{o}_t) = \frac{\exp\left(\mathbf{o}_{tj}^{N}\right)}{\sum_{\mathbf{s}} \exp\left(\mathbf{o}_{t\mathbf{s}}^{N}\right)} \approx \frac{\exp\left(\mathbf{o}_{tj}^{N}\right)}{\sum_{k \in n-\mathrm{best}} \exp\left(\mathbf{o}_{tk}^{N}\right)} \tag{16}$$

Where exp $O_{tj}^N \approx 0$ for $j \notin n-$best list.

4 Experiments

In this section, the standard TIMIT phone recognition task is used to evaluate the proposed approach described in this paper. The training sets consist of 462 speakers and results are computed using the 24 speaker core test set. The DNN training development set is based on 50 speakers from the test set [22]. The SA1 and SA2 utterances were not used.

The speech was analyzed using a 25ms Hamming window with a 10 ms fixed frame rate. The speech is represented using 12 mel frequency cepstral coefficients (MFCCs), energy, along with their first and second temporal derivatives, resulting in a 39 element feature vector. Another representation used for DCRFs is based on using a Fourier-transform-based filter-bank with 40 coefficients (plus energy) distributed on a mel-scale, together with their first and second temporal derivatives resulting in a 123 element feature vector. The features are pre-processed to have zero mean and unit variance and acoustic context information is integrated using a window of 9 frames (4 left + current frame+ 4 right) to construct the final frames.

The original 61 phone classes in TIMIT were mapped to a set of 48 labels, which were used for training [23]. After decoding, this set of 48 phone classes

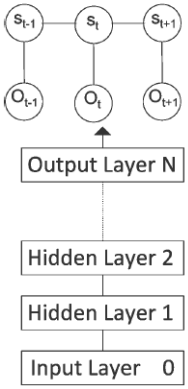


Figure 3: Linear chain DCRF model (the state scores are computed from a DNN).

was mapped down to a set of 39 classes. The phone error rate (PER) metric, which is analogous to word error rate, is used to report phone recognition results. Each phone of the baseline HMMs was represented using a three state left-to-right model. Mixtures of Gaussian densities with diagonal covariance matrices were used for state scoring (emission probabilities). The HMMs were trained by the maximum likelihood criterion using the conventional EM algorithm [24]. Discriminative training based on Minimum Phone Error (MPE) criterion was used to refine the HMMs [25]. The acoustic scale was set to 1/6 and I-smoothing parameter τ was set to 100.

Similar to the HMMs, the ACRF-based models emply three-state left-to-right phone models. The transition parameters were initialized from trained HMM models. Other parameters were initialized to zero. The same model structure was used for postprocessor ACRFs. A Viterbi pass (forced alignment) of the reference transcription using HMMs trained using the maximum likelihood criterion was used to accumulate the M^{num} sufficient statistics. The number of frames in the acoustic context window, w = 2c + 1, was set to 19. For ACRFs primary acoustic models, 7917 Gaussians were used to construct the augmented space. A powerful primary acoustic model based on DCRFs was evaluated. Each phone

was represented using a three state left-to-right DCRF. The transition parameters were initialized from trained HMM models as in ACRFs. The DNN parameters were initialized to random values. The DNN has nine hidden layers and each layer has 2048 neurons. For training DCRFs, the PDNNTK toolkit [26] in combination with the Theano library [27] is used, allowing transparent

Table 1: HMM decoding results on TIMIT recognition task in terms of PER

Model	10 Mix	40 Mix
HMM baseline	32.3%	29.9%
ACRF postprocessor1	28.7%	27.9%
ACRF postprocessor2	28.2%	27.5%

Table 2: Decoding results on TIMIT recognition task in terms of PER for different primary acoustic models.

Model	ACRFs	DCRFs
baseline	27.3%	22.7%
ACRF postprocessor1	26.7%	22.3%
ACRF postprocessor2	26.6%	22.5%

Computation for CPUs and GPUs.

The acoustic modeling process starts with generating the state scores of the primary models in pruned posterior forms. These scores are fed to the first ACRF postprocessor. The output state scores of the firrst ACRF post processor are generated in pruned posterior forms and are fed to the second ACRF postprocessor in all experiments. A generic bi-gram in-house decoder is used to generate the recognition phone sequence for the different acoustic models. Table 1 shows the decoding results when HMMs are used as a primary acoustic model. The results show that the first stage of ACRFs post processing leads to significant improvement in terms of PER. When ACRFs and DCRFs were used as primary acoustic models, the improvements are smaller than HMMs as shown in Table 2. The second stage of post processing did not lead to improvements. These results may suggest that ACRF post processing has limited ability for powerful acoustic models.

5 Conclusions

In this paper, an augmented conditional random field postprocessor for speech recognition is presented. In this framework, a primary acoustic model is used to generate state posterior scores per frame. These posterior scores are then used as input to an ACRF. The main goal of this process is to model the acoustic context information in a high dimensional space constructed using the primary acoustic model state scores. Consequently, the postprocessor acoustic model discovers new context information and improves the recognition accuracy. Three different primary acoustic models were investigated in this work (HMM, ACRF, and DCRF). Results on the TIMIT phone recognition task show that the proposed postprocessor can lead to significant improvements especially when HMMs were used as a primary acoustic model.

REFERENCES

[1] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: Proc. ICML,2001, pp. 282{289.

[2] M. Layton, M. Gales, Augmented statistical models for speech recognition, in: Proc. IEEE ICASSP, Vol. 1, France, 2006, pp. 129{ 132.

[3] J. Morris, E. Fosler-Lussier, Conditional random fields for integrating local discriminative classifiers, Audio, Speech, and Language Processing, IEEE Transactions on 16 (3) (2008) 617{628. doi:10.1109/TASL.2008.916057.

[4] G. Zweig, P. Nguyen, D. V. Compernolle, K. Demuynck, L. Atlas, P. Clark, G. Sell, M. Wang, F. Sha, H. Hermansky, D. Karakos, A. Jansen, S. Thomas, G. Sivaram, S. Bowman, J. Kao, Speech recognition with segmental conditional random fields: A summary of the JHU CLSP summer workshop, in: Proc. IEEE ICASSP, 2011.

[5] M. Gales, S.Watanabe, E. Fosler-Lussier, Structured discriminative models for speech recognition, IEEE Signal Processing Magazine.

[6] Y. Hifny, Conditional random fields for continuous speech recognition, Ph.D. thesis, University Of Sheffield (2006).

[7] Y. Hifny, S. Renals, Speech recognition using augmented conditional random fields, IEEE Transactions on Audio, Speech and Language Processing 17 (2) (2009) 354{365.

[8] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society 39 (1) (1977) 1{38.

[9] L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. of IEEE 77 (2) (1989) 257{286.

[10] F. Jelinek, Statistical Methods for Speech Recognition, MIT Press, 1997.

[11] X. Huang, A. Acero, H.-W. Hon, Spoken Language Processing: A Guide to Theory, Algorithm and System Development, Prentice Hall, 2001.

[12] J. Bilmes, What HMMs can do, IEICE Transactions on Information and Systems E89-D (3) (2006) 869{891.

[13] Y. Hifny, Acoustic modeling based on deep conditional random fields, Deep Learning for Audio, Speech and Language Processing, ICML.

[14] S. Renals, N. Morgan, H. Bourlard, M. Cohen, H. Franco, Connectionist probability estimators in HMM speech recognition, IEEE Transactions on Speech and Audio Processing.

[15] N. Morgan, H. Bourlard, Continuous speech recognition: An introduction to the hybrid HMM/connectionist approach, IEEE Signal Processing Magazine 12 (3) (1995) 25{42.

[16] B. Kingsbury, Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling, in: Proc. IEEE ICASSP, 2009, pp.3761{3764. doi:10.1109/ICASSP.2009.4960445.

[17] R. Prabhavalkar, E. Fosler-Lussier, Backpropagation training for multilayer conditional random _eld based phone recognition, in: Proc. IEEE ICASSP, Vol. 1, France, 2010, pp. 5534 { 5537.

[18] G. Hinton, L. Deng, D. Yu, G. Dahl, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, , B. Kingsbury, Deep Neural Networks for acoustic modeling in speech recognition, IEEE Signal Processing Magazine.

[19] E. T. Jaynes, On the rationale of maximum-entropy methods, Proc. Of IEEE 70 (9) (1982) 939{952.

[20] Y. Hifny, S. Renals, N. Lawrence, A hybrid MaxEnt/HMM based ASR system, in: Proc. INTERSPEECH, Lisbon, Portugal, 2005, pp. 3017{3020.

[21] A. Mohamed, D. Yu, L. Deng, Investigation of full-sequence training of Deep Belief Networks for speech recognition, in: Interspeech, 2010.

[22] A. Halberstadt, J. Glass, Heterogeneous measurements and multiple classifiers for speech recognition, in: Proc. ICSLP, Vol. 3, Sydney, Australia, 1998, pp. 995{998.

[23] K.-F. Lee, H.-W. Hon, Speaker-independent phone recognition using hid-den Markov models, IEEE Transactions on Speech and Audio Processing 37 (11) (1989) 1641{1648.

[24] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, P. Woodland, The HTK Book, Version 3.1, 2001.

[25] D. Povey, Discriminative training for large vocabulary speech recognition, Ph.D. thesis, University of Cambridge (2004).

[26] Y. Miao, PDNN: Yet Another Python Toolkit for Deep Neural Networks. URL http://www.cs.cmu.edu/ ymiao/pdnntk.html

[27] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Des-jardins, J. Turian, D. Warde-Farley, Y. Bengio, Theano: a CPU and GPU math expression compiler, in: Proceedings of the Python for Scientific Computing Conference (SciPy), 2010, oral Presentation.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
VOLUME 3, ISSUE 2
ISSN 2054-7390

English Premier League (EPL) Soccer Matches Prediction using An Adaptive Neuro-Fuzzy Inference System (ANFIS) for

Amadin, F. I[1] and Obi, J.C.[2]
Department of Computer Science, University of Benin, Benin City. Nigeria.
frankamadin@uniben.edu[1]; tripplejo2k2@yahoo.com[2]

ABSTRACT

Prediction of English Premiership League (EPL) matches has been on the heart and minds of researcher over the pass decades, but none has sufficiently introduced and Adaptive Neuro-Fuzzy Inference System (ANFIS) approach for these prediction which has served as the focal aim of this research paper using seven premier league teams and nine linguistic values. Matric Laboratory (MATLAB) 7.0 served as the tool of implementation highlighting various views. The ANFIS training was successful completed at epoch 2 and having an error of 1.41237e-006. The model was further used to predict the outcome of 7 matches with a successful rate of 71%.

Keywords: ANFIS, Premiership, Soccer, Predication, Layers

1 Introduction

It is no doubt that the most viewed sport in the world is soccer having above 10 billon fan all over the world (Tony, 2014). The English league is the most watched soccer league all over the world having more than 2 billion fans (Jonathan, 2014 and Tony, 2014). It comprises of 20 teams and each team plays 38 match in a season which spans ten months. A team plays 19 matches at home (at the city in which the club originates) and 19 match away (Jonathan, 2014). A win for any match regardless of home or away is 3 points a draw is 1 point while a loss earn no point. At the end of each season the 3 teams at the lower end (bottom) of the league are relegated to a lower division (Tony, 2014). The outcome of any match is a win, a draw or a loss (Jonathan, 2014 and Tony, 2014). To calculate the number of games played in a season using the hand shaking lemma we represent each team as a node and a matched played as an edge. There are two parallel edges between each node (one for home match and the other for away match). So, the number of game played = (38*20) / 2 =380 matches. Since the edge contribute twice to the degree of the graph.

2 Review of Related Literature

In predicting the outcome of a soccer match, various approaches have been used some of which include Statistical method, Probabilistic method, Bayesian network, Multilayer perceptrons.

Yue (2003) used multilayer perceptron model to predict the outcome of a soccer match. The neural network had 13 inputs, 3 hidden layers and 1 output layer. The network was feed with the most recent 5 matches of team A and team B. The model was used to predict the outcome of 9 matches and 4 out of the nine matches gave accurate prediction. Based on the efficiency of the result, the model was re-

DOI: 10.14738/tmlai.32.1027
Publication Date: 28th April, 2015
URL: http://dx.doi.org/10.14738/tmlai.32.1027

implemented with two additional hidden layers (a layer for the home team with a weight of 1 and the other for the position of the team on the table of ranking) which was utilized in predicting nine matches, with an 80% accuracy Aditya et al (2013) used a Multi Nominal Logic Regression (MNLR) and Support Vector Machines (SVM) to predict the outcome of a match utilizing 3 fundamental approaches. In their first approach Multi nominal Logistic was enacted for the training phase in which performance optimization metrics derived from current matches were considered, rather than taking the average of previous matches and during testing they predicted the outcome between two teams using the number of matches past played. In the second approach, they trained in the same way they later test the data and instead of using the feature vector as the performance metric vector corresponding to the current match, they used Key Performance Parameters (KPP). The third approach tried to find a global set of parameter which was independent of the competing teams.

Ian and Phil (2013), forecasted international; soccer matches result using bivariate discrete distribution. They collected a total of 8,735 international soccer results from two main sources. The data for the period 1993-2001 was obtained from the archive of International Soccer Results (ISR) and the data for the period 2001 to 2004 were obtained from the Record Sport Soccer Statistics Foundation (RSSSF) archive. Data on the Federation of International Football Association (FIFA) world rankings were collected from the FIFA website for each month during the years from 1993 to 2004. The model was based on copula functions. The copula regression model forecasts 41.8% of the results correctly.

From the review of related literature An Adaptive Neuro-Fuzzy Inference System (ANFIS) approach has not been adopted previously for EPL matches prediction which is serving as the focal point of our research.

3 The ANFIS Model for Predicting the Outcome of English Premiership League (EPL) Soccer Match

The Adaptive Neuro Fuzzy Inference System (ANFIS) is one of the many hybrids of neural and fuzzy system. ANFIS combines the learning capability of the neural network and the explanatory power of the fuzzy system. The ANFIS architecture uses the sugeno type inference system Jang (1991). An example of the sugeno inference is given in equation 1:

$$\text{If } x \text{ is } A \text{ and } y \text{ is } B, \text{ then } f1 = p1x + q1y + r1 \quad (1)$$

Where x and y is the input variable and A and B is the fuzzy set of linguistic variables and q, p and r, are consequent parameters.

ANFIS ARCHITECTURE

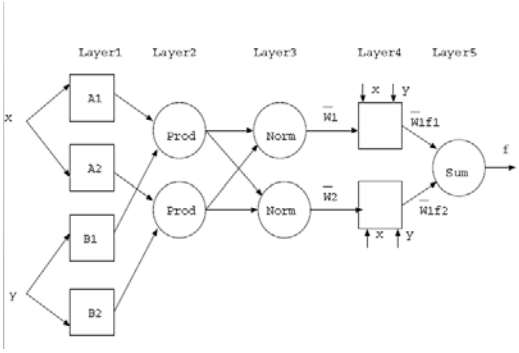


Figure 1: ANFIS ARCHITECTURE

The ANFIS architecture in Figure 1 comprises of varied layers which includes:

a. ***Layer 1 (Input Layer)***: this is the first layer it is called the input layer. Each node generates the membership grades of a linguistic label (Jang, 1993). An example of a membership function is the generalized bell function is given in equation 2:

$$\mu_{A_i}(x) = \frac{1}{1+[\frac{(x-c_i)^2}{a_i^2}]^{b_i}} \tag{2}$$

Where {a, b, c} is the parameter set and x stands for the individual value. As the values of the parameters change, the shape of the bell-shaped function varies. Parameters in that layer are called premise parameters.

b. ***Layer 2 (Fuzzification layer):*** The second layer of the Adaptive Neuro Fuzzy Inference System (ANFIS) model is fuzzification layer it is also called the membership function layer. In this layer the output of the first layer is mapped to a fuzzy set using bell-shaped membership function (Jang, 1997 and Nauck et al., 1997). The Bell membership function provides smooth and non-linear functions that can be used by the learning systems. Each node calculates the firing strength of each rule using the min or prod operator show in equation 3. In general, any other fuzzy AND operation can be used.

$$O_{2,i} = w_i = \mu_{A_i}(x) \times \mu_{B_i}(x) \qquad i = 1,\ 2 \tag{3}$$

c. ***Layer 3 (Rule Layer):*** The third layer is the rule layer. Each neuron in this layer corresponds to a single first-order Sugeno fuzzy rule receiving signal from the membership function layer and computes the truth value of the rule (Jang, 1993 and 1997. The nodes calculate the ratios of the rule's firing strength to the sum of all the rules firing strength using the equation 4. The result is a normalised firing strength.

$$O_{3,i} = \overline{w}_i = \frac{w_i}{w_1 + w_2} \qquad i = 1,\ 2 \tag{4}$$

d. ***Layer 4 (Normalization layer):*** The fourth layer is the normalization layer. Each neuron in this layer receives signals from all rule neurons in the third layer, and calculates the normalized firing strength of a given rule. The nodes compute a parameter function on the layer 3 output using equation 5. Parameters in this layer are called consequent parameters.

$$O_{4,i} = \overline{w}_i f_i = \overline{w}_i (p_i x + q_i y + r_i) \tag{5}$$

e. ***Layer 5 (Defuzzification layer)***: The fifth layer is the defuzzification layer. Each neuron in this layer is connected to the respective normalization neuron in the fourth layer (Jang, 1997 and Nagnevitsky, 2002). It is a single node that aggregates the overall output as the summation of all incoming signals using equation 6

$$O_{5,1} = overall\ output = \sum_i \overline{w}_i f_i \tag{6}$$

3.1 Generating Dataset

In generating the dataset we relied on 5 factors which then constituted the ANFIS parameters which was then used to predict the outcome of the match, they are:

- 2 of the last most recent match played by team A
- 2 of the last most recent match played by team B
- The point of both teams in the table of ranking
- Their popularity
- Home advantage.

The first 2 parameters (2 of the last most recent match played by team have 9 linguistic variable (WW, WD, WL, DW, DD, DL, LW, LD, LL) while the point of both teams and the popularity of the teams have 3 parameters (highA, same, lowA) and the home advantage have 2 linguistic variable (homeA and awayA).

3.2 Variable Used In the Model

Table 1, shows, six linguistic variables and six linguistic values associated with the Anfis proposed architecture for predicting the EPL results and matches

Table 1: Fuzzy Linguistic Variables and Values

Position	Club	Last Two Match Played	Point
1	Chelsea	WD	33
2	Manchester City	WW	25
3	Southampton	WW	22
4	Manchester United	WW	21
5	West ham United	DW	20
6	Arsenal	LW	20

4 Stimulations and Experiment

The stimulation was carried out using MATLAB Fuzzy logic toolbox 2007 and the model was used to predict the outcome of 7 matches. The various simulation modules are specified from Figure 2, 3, 4 and 5 respectively

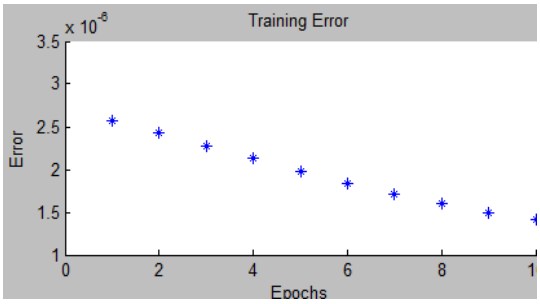


Figure 2: Training Error

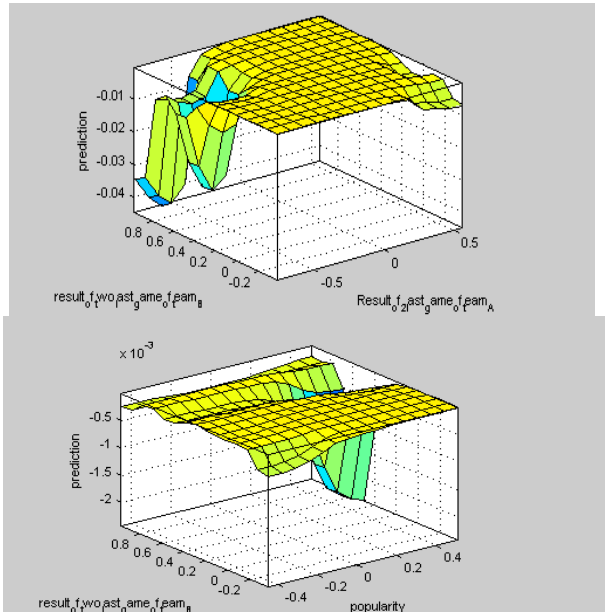


Figure 3: Surface View Prediction 1

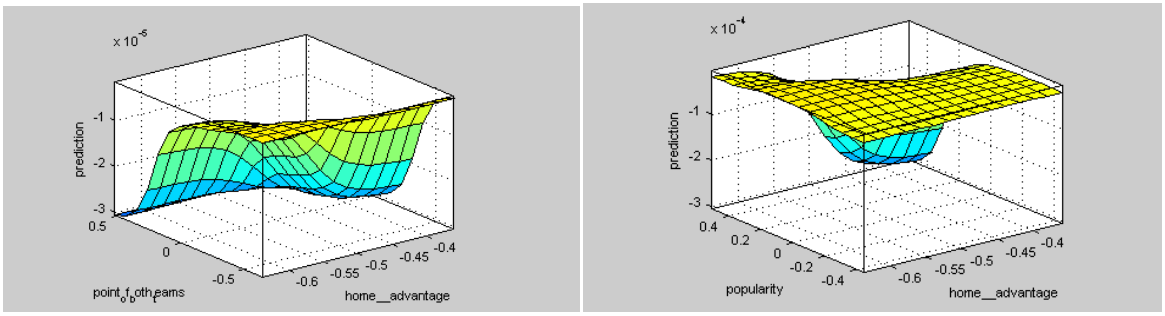


Figure 4: Surface View Prediction 2

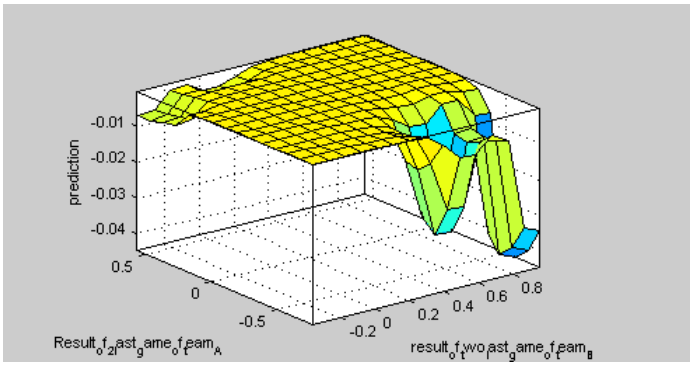


Figure 5: Surface View Prediction

The ANFIS training was completed at epoch 2 and having an error of 1.41237e-006. The model was further used to predict the outcome of 7 matches and 5 outcomes out of the matches were accurately predicted.

4.1 Prediction Results

The EPL prediction matches are specified clearly on Table 2, highlighting both actual result and prediction results

Table 2: English Premier League (EPL) Prediction Matches

Home Team	Away Team	Actual Result	Prediction
Manchester	Stoke	Manchester	Manchester
Chelsea	Tottenham	Draw	Chelsea
Sunderland	Man City	Man City	Man City
Everton	Hall City	Draw	Draw
Arsenal	Southampton	Arsenal	Southampton
Leister City	Liverpool	Liverpool	Liverpool
Swansea	QPR	Swansea	Swansea

5 Conclusion

The ANFIS model was able to make an accurate prediction of 5 out of 7 matches which is promising but this implies that there might be more factors which were not included in our model that determine the outcome of a soccer match. So, further study is encouraged to increase the accuracy of prediction of a soccer match suing more applicable variables.

REFERENCES

[1] Aditya S. T., Aditya P. and Vikesh K. (2013)" Game ON! Predicting English Premier League Match Outcomes" retrieved from gameON.com

[2] Ian M. and Phil S (2013), "Forecasting international soccer match results using bivariate discrete distributions" Centre for Operational Research and Applied Statistics, Salford Business School, University of Salford, Salford, Manchester UK.

[3] Jang, R (1991). "Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm". Proceedings of the 9th National Conference on Artificial Intelligence, Anaheim, CA, USA, pp. 762–767. Retrieved on October 15th 2014 from www.irrolecom/article/00098.pdf

[4] Jang, J. (1993). "ANFIS: adaptive-network-based fuzzy inference system". IEEE Transactions on Systems, Man and Cybernetics Volume23 issue3. doi:10.1109/21.256541.

[5] Jang, S (1997) "Neuro-Fuzzy and Soft Computing "Prentice Hall, Pp 335–368, ISBN 0-13-261066-3

[6] Jonathan T. (2014), "2014-15 English Premier League and MLS TV"rtrived online from http://www.philly.com/philly/blogs/thegoalkeeper/NBC-Sports-2014-Premier-League-TV-schedule-for-August-through-November.html#sd6Wr3imqE4ehOdH.99

[7] Nauck, D., Klawon F. and R. Kruse, (1997), "Foundations of Neuro-Fuzzy Systems", J. Wiley & Sons pages 312

[8] Negnevitsky, M., (2002) "Artificial Intelligence: A Guide to Intelligent Systems", 2d ed. Harlow, England: Addison Wesley, pp. 90-343.

[9] Tony M. (2014), "20 English Premiership Teams" http://www.businessinsider.com/new-premier-league-kits-2014-8?op=1

[10] Yue W. M. (2003)" Prediction on Soccer Matches using Multi-Layer Perceptron" ID: 903-051-7735

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
VOLUME 3, ISSUE 2
ISSN 2054-7390

Authorship Identification using Generalized Features and Analysis of Computational Method

[1]Smita Nirkhi, [2]R.V.Dharaskar and [3]V.M.Thakare
[1]G.H.Raisoni College of Engineering, Nagpur University, Nagpur, India;
[2]Disha Technical Campus, Raipur, India;
[3]Department of Computer Science, University Campus, Amravati, India
smita811@gmail.com; rvdharaskar@yahoo.com; vilthakare@yahoo.co.in

ABSTRACT

Authorship Identification is being used for forensics analysis and humanities to identify the author of anonymous text used for communication. Authorship Identification can be achieved by selecting the textual features or writing style. Textual features are the important elements for Authorship Identification .It is therefore important to analyze them and identify the most promising features. This paper tries to identify and analyze promising generalized features and computational methods for authorship Identification. The performed experiments in the authorship identification task shows that, the support vector machine classifier used as computational method can achieve better results with identified generalized feature set.

Keywords: Author identification, support vector machine, feature extraction, classification

1 Introduction

Internet has provided us a platform and convenient way to share information across time and place. At the same time it is also used for criminal activities like Cyberattacks, Distribution of illegal materials in cyberspace, Computer-mediated illegal communications within big crime groups or terrorists. Cybercrime has become one of the major securities Issues for the law enforcement community. The anonymity of cyberspace makes identity tracing a significant problem which hinders investigations. Anonymity means senders will attempt to hide their true identities to void detection. Cybercriminals also Forged sender's address and Use multiple usernames to distribute online messages via different anonymous channels. Cybercrimes due to anonymity includes 1.Identity theft and masquerade 2.Phishing and spamming 3.Child pornography 4.Drug trafficking 5.Terrorism 6.Infrastructure crimes: Denial of service attacks. The possible solution for above mentioned problem is identifying the writing style of these messages. Cyber-criminal may have "word print" hidden in his online messages.

This study proposes the use of authorship analysis approach to solve the problem of identity tracing in cybercrime investigation. Problem statement is to verify whether suspect S is or is not the author of a given malicious e-mail or online message µ.with assumption that investigator has access to previously written e-mails of suspect S and have access to e-mails {E1,…,En}, collected from sample population U= {u1,…,un}.The task is to extract stylometric features and develop two models that is Training model & testing model. After that classify e-mail µ using the two models. Feature selection and computational methods are two critical research issues that influence the performance of authorship analysis. Selected

DOI: 10.14738/tmlai.32.1064
Publication Date: 28th April, 2015
URL: http://dx.doi.org/10.14738/tmlai.32.1064

features should be effective discriminators. Computational Methods provides approach to discriminating texts by authors based on the selected features. Next section of paper describes the existing work in the authorship identification field by analyzing the various features used in various research papers along with their accuracy followed by another section experimental setup which is describes the methodology used for performing experimentation. Last section shows the experimental results in terms of accuracy.

2 Related Work

Authorship analysis is categorized into three major categories [11]

1. Authorship identification (authorship attribution) which determines the likelihood of a piece of writing to be produced by a particular author by examining other writings by that author
2. Authorship characterization:-It summarizes the characteristics of an author and generates the author profile based on his/her writings along with Gender, educational, cultural background, and writing style
3. Similarity detection:-It Compare multiple pieces of writing and determines whether they were produced by a single author without actually identifying the author for e.g. Plagiarism detection.

In previous work Writing-style Features applied for Authorship Identification are Lexical features, syntactic features, Structural features & Content-specific features. Lexical features (F1) based on words and character analysis. Syntactic features (F2) perform function words, punctuation usage; POS. Structural features (F3) make use of signature, personal article-organizing style. Content-specific features (F4) analyze consistently used and content-related key words [11].Various researchers used these features for experimentation and accuracy is calculated. The table-1 shows the features and technique used for Authorship Identification along with accuracy. Chaski (2005)[8] has achieved 95.70% accuracy by using feature set (F1,F2,F3,F4) and 10 authors were used for experimentation. Similarly Iqbal (2008)[9] used Frequent Pattern Mining Algorithm to extract writing style of author. Hadjidj (2009)[7] used F1,F2,F3,F4 with accuracy 90%. Iqbal (2010)[10] used K-means with accuracy 90%. Zheng used F1, F2, F3, F4 and achieved 97.69% accuracy.

Table 1: Features used in various Research Paper

Research Paper	Number Of Authors	Features /Technique Used	Accuracy	Number Of Authors
Chaski(2005)[8]	10	F1,F2,F3,F4	95.70%	10
Iqbal(2008)[9]	10	Frequent Pattern Mining	77%	10
Hadjidj(2009)[7]	3	F1,F2,F3,F4	90%	3
Iqbal(2010)[10]	3	K-means	90%	3
Zheng	10	F1,F2,F3,F4	97.69%	10

Performance for Authorship Identification can be measure in terms of Accuracy and number of Authors used for analysis.Table-2 shows previous work done in terms of number of authors used for experimentation.

Table 2: Experimental setups from previous research.

Research Paper	Total Number of persons(P)	Total Number Of messages	Average message Length(Word)	Average Message per person
Corney et al	4	253	92	64
De Vel	3	156	259	52
Zheng et al	20	960	169	48
Stamatotes	10	300	1122	30
Tsuboi	3	4961	112	1653

From above literature survey it is observed that despite significant progress achieved on the identification of an author within a small group of individuals, it is still challenging to identify an author when the number of candidates increases .secondly, it is difficult to identify if the sample text is short as in the case of e-mails or online messages. The following experimentation shows that for short text and more number of authors, proposed methods gives more accuracy using new feature set.

3 Experimental Setup

Figure 1 shows computational methods that can be used for experimentation are divided into two categories. Those are

1. Statistical Approach uses cluster analysis and Multidimensional Scaling.
2. Machine Learning Approach uses SVM, Naïve Bays

In this paper, For Experimentation purpose method used is Support Vector Machine and Features used are most frequent words means the words with highest frequency are considered in analysis and n-gram approach.

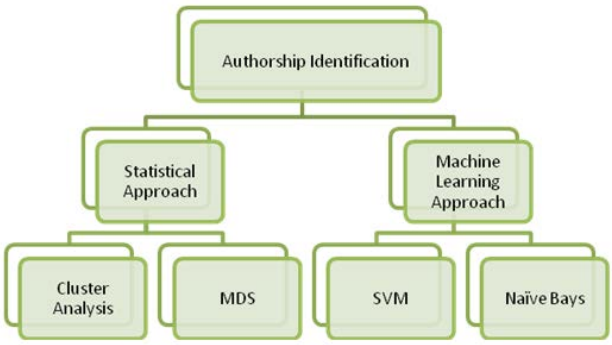


Figure 1: Computational Methods

3.1 Corpus used

3.1.1 C50 corpus

The C50 dataset was downloaded from the UCI Machine Learning Repository. It consists of one training and one test set, these sets are not overlapping. Each of the datasets contains 2500 documents (50 authors with 50 documents each) in text format. All of the documents are written in English and belongs to the same subtopic which will minimize the possibility of being able to classify documents depending on topics instead of the unique features which represent each author.

3.1.2 Enron corpus

Enron corpus was made public during the legal investigation concerning the Enron Corporation. The current version contains 619,446 messages belonging to 158 users

This dataset was collected and prepared by the CALO This data was originally made public, and posted to the web, by the Federal Energy Regulatory Commission during its investigation.

4 Experimental Results

Table3 shows experimental results on C50 dataset with number of authors =5, 7,15,25,50 using SVM classifier and n-gram for word=1

Table 3: Experimentation on c50 Dataset

Data set Used	Total Number Of Authors	Training set	Testing set	Accuracy
C50 Dataset	5	173	173	82.5%
	7	173	2	100%
	15	373	3	86.5%
	25	625	2	100%
	50	625	10	88%

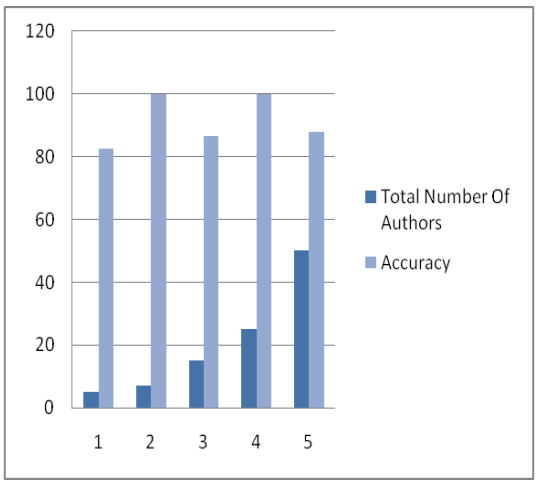
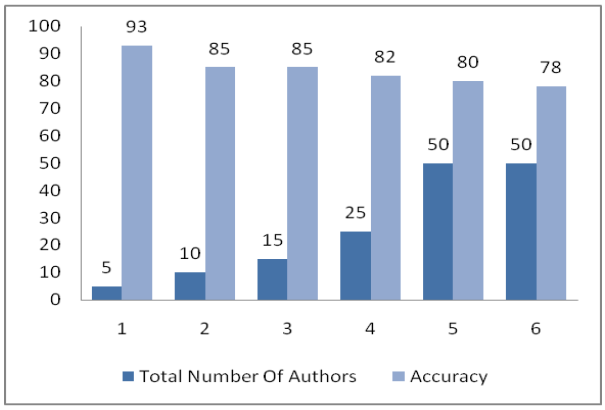


Table 4 shows experimental results on Enron dataset with number of authors =5, 10,15,25,50 using SVM classifier and n-gram for word=1

Table 4: Experimentation on Enron Dataset

Data set Used	Total Number Of Authors	Training set	Testing set	Accuracy
Enron Dataset	5	62	10	93.3%
	10	125	10	85%
	15	188	10	85%
	25	310	13	82%
	50	650	2	80%



5 Conclusion

The proposed approach is able to identify the authors of online messages. Character and word Uni-gram features showed particular discriminating capabilities for authorship identification. SVM gives more accuracy with word uni-gram .Different parameter settings of authorship identification had an impact on performance. The above experimentation shows that for short text and more number of authors, proposed methods gives more accuracy using n-gram approach for feature set.

REFERENCES

[1] Abbasi, A., & Chen, H. (2005). Analysis to Extremist- Messages, (October), 67–75.

[2] B. Loader, D.Thomas (Eds), Cybercrime: Law enforcement, security and surveillance in the information age. Routledge; 2000.

[3] A. Abbasi, H. Chen. "Writeprint: A stylometric approach to identity level identification and similarity detection in cyberspace". ACM Transaction on Information System, 26(2):1-29, 2008

[4] R. Zheng, J. Li, H. Chen, Z. Huang. "A framework for authorship identification of online messages: Writing-style features and classification techniques". Journal of the American Society for Information Science and Technology, 57(3), pp.378-393, 2006.

[5] S. Nizamani S, N. Memon N, U. K. Wiil, P. Karampelas, "CCM: A Text Classification Model by Clustering", International Conference on Advances in Social Networks Analysis and Mining (ASONAM). Kaohsiung, Taiwan, pp.461-467, 2011.

[6] UCI Machine Learning Repositiory, Reuter 50 50 Dataset. https://archive.ics.uci.edu/ml/datasets/Reuter_50_50.

[7] R. Hadjidj, M. Debbabi, H. Lounis, F. Iqbal, A. Szporer, and D. Benredjem. Towards an integrated e-mail forensic analysis framework. Digital Investigation, 5(3-4):124 – 137, 2009

[8] C. E. Chaski. Who's at the keyboard: Authorship attribution in digital evidence Investigations International Journal of Digital Evidence, 4(1), Spring 2005.

[9] F. Iqbal, R. Hadjidj, B. C. Fung, and M. Debbabi. A novel approach of mining write-prints for authorship attribution in e-mail forensics. Digital Investigation, 5, Supplement (0):S42 – S51, 2008. The Proceedings of the Eighth Annual DFRWS Conference

[10] F. Iqbal, H. Binsalleeh, B. C. Fung, and M. Debbabi. Mining writeprints from anonymous e-mails for forensic investigation. Digital Investigation, 7(1-2):56 – 64, 2010.

[11] S.M.Nirkhi, R. V. Dharaskar, V.M.Thakre, "Analysis of online messages for identity tracing in cybercrime investigation", 2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec), pp. 300 - 305, 2012

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
VOLUME 3, ISSUE 2
ISSN 2054-7390

Mobile Agent Life Cycle Demystified using Formal Method

IMIANVAN Anthony Agboizebeta[1] and AKINYOKUN Oluwole Charles[2]
[1]*Department of Computer Science, University of Benin, Benin City. Nigeria*
[2]*Department of Computer Science, Federal University of Technology, Akure, Nigeria.*
tonyvanni@uniben.edu

ABSTRACT

Underlying technique for mobile agent development is often mystified. Existing research sometimes ignore unveiling the details of the mobility and autonomy of the agent system. This paper exposes using formal methods the technique of a mobile agent system using a life cycle. The system proposed will serve as takeoff springboard for mobile agent developers.

Keywords: Mobile Agent, Formal Method, Z-notations, Mobile Agent Life Cycle.

1 Introduction

Mobile agents are autonomous and intelligent programs that are capable of moving through a network, searching for and interacting with the resources on behalf of the network administrator. Mobile agent is an executive program that can migrate at times of its own choosing from one machine to another in a network. This means that a mobile agent is 'free' to travel to any place in the network. It can execute without requiring a link with or being controlled from the originating location (Imianvan, 2009).

Mobile agent could be activated and launched from one computer to another for the purpose of autonomously searching for and interacting with network resources on the network administrator's behalf (Imianvan and Akinyokun 2014). A conscious effort at developing a mobile agent for the assessment and evaluation of computer networks with emphasis on throughput, utilization and availability have been proposed in (Aderounmu, 2001). Mobile agent technology is useful in electronic commerce transactions, distributed information retrieval, and network management (Huy et al., 2005; Imianvan, 2009; Weina and Gaoyuan 2011; Djamel et al., 2012).

The following are examples of some existing mobile agent systems. The items inscribed in the brackets are the individual, or institution or organizations that developed them.

a. Agentspace (Alberto Silva).
b. Agent TCL (Dartmouth).
c. Aglets (IBM).
d. D' Agents (Dartmouth College).
e. Dicovery (University of Maryland).
f. JATLite (Standford University).
g. MARS (University of Modena).
h. Messengers (University of California).
i. Mobile Agent Platform (Universita' di Catania).
j. TACOMA: Tromso and Cornell Moving agent (University of Tromso).

DOI: 10.14738/tmlai.32.1025
Publication Date: 28th April, 2015
URL: http://dx.doi.org/10.14738/tmlai.32.1025

Mobile agent like every other intelligent agent has the following features:

a. Autonomy.
b. Adaptiveness.
c. Collaborative behavior.
d. Character.

A specification written and approved in accordance with established (mathematical) notations is a formal specification. Z ('zed'), for instance is a formal notation based on set algebra and predicate calculus for the specification of computing systems. Z specification of systems employs the power of discrete mathematics. The Z notation is useful to organize and communicate thoughts within a design team (Diller 1994; Spivey 1998).

Since a formal specification is precise, if such a specification is wrong, it is easier to tell where it is wrong and correct it. Using formal notation increases the understanding of the operation of a system especially early in a design. It helps to organize the thoughts of a designer, making clearer, simpler designs possible. Formal specification provides a check that the system will behave as expected by the designer. The use of formal methods can help to explore design choices. Such methods aid the design team in reasoning about the operations of the system in clear terms before and during its implementation (Spivey, 1998).

This paper provides an attempt to demystify using formal methods (Zed notations), the operational life cycle of a mobile agent system.

2 The Mobile Agent Life Cycle

A mobile agent is an artificial life which is capable of birth (creation), survival (launching) and death (disposal). The processes of birth, survival and death are characterized by a sequence of logical steps called the mobile agent life cycle as presented in Figure 1.

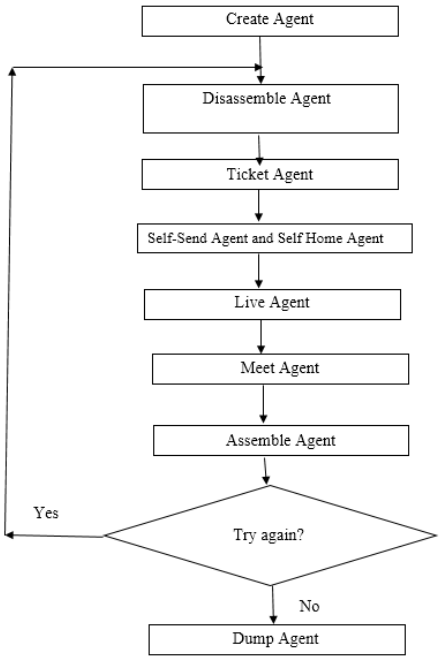


Figure 1: The Mobile Agent Life Cycle

The mobile agent life cycle presented in Figure 1 involve the following series of logical procedures (Imianvan, 2009).

a. CreateAgent.
b. DisassembleAgent.
c. TicketAgent.
d. SelfSendAgent and SelfHomeAgent.
e. LiveAgent.
f. MeetAgent.
g. AssembleAgent.
h. DumpAgent.

The creation of the mobile agent involves developing the functionality of the system and then adding it to the universal set of agent as presented in Figure 2.

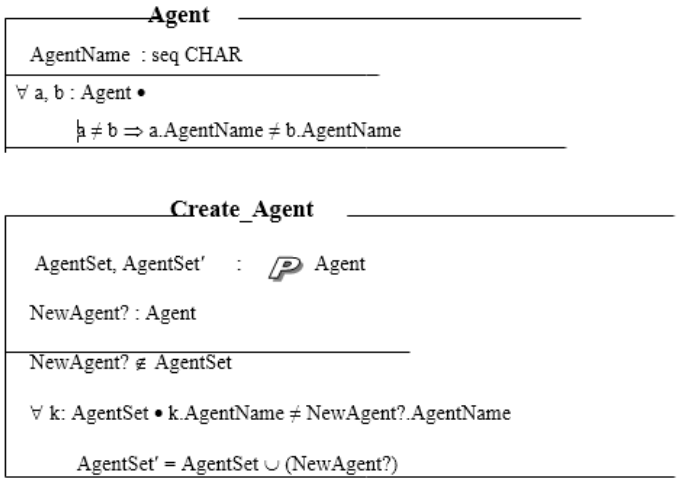


Figure 2: Formal Specification of the CREATE Agent Process

At the source, the mobile agent is decomposed into units that are transportable to the target workstations. The formal specification of the disassembling of the mobile agent is presented in the Z Schema of Figure 3.

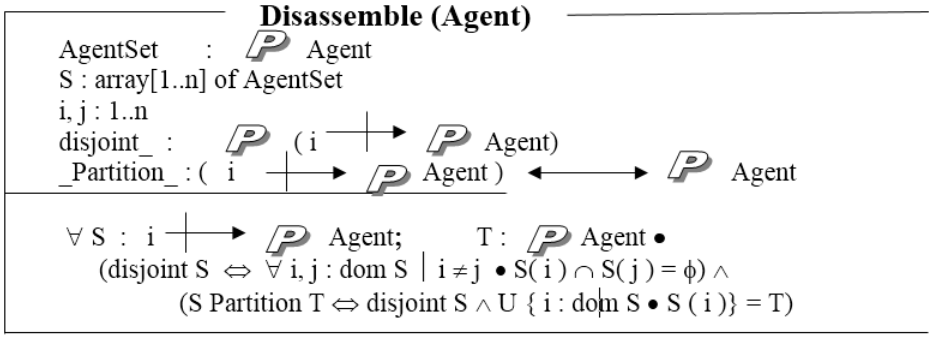


Figure 3: Z Schema Specification for Mobile Agent Disassembling.

Operational procedure of TicketAgent component of mobile agent system is presented in Figure 4.

```
1.  TicketAgent(agent, v, e)
2.      // agent is to be deployed to workstations.
3.      // v is set of vertices of the graph representing workstations.
4.      // e is set of edges of the graph representing distance between workstations.
5.      TargetFound ← false
6.      j ← 1
7.      loop i from 1 to number of workstations
8.          TeleAddress wi      // tag target addresses.
9.          LookUpPlaces wi     // used to autonomously view targets workstations.
10.         //  wi is target workstations
11.         if wi  not in v
12.             v ←  v U {wi}
13.         end if
14.         copy(v)     // collect or copy target address
15.         j ← j + 1
16.         if wi ≠ wj edgei,j  ← distance wj – distance wi
17.             e   ←  e U edgei,j
18.         end if
19.         copy(e)   // collect or copy distance between targets
20.         TargetFound  ← true
21.         Ticket (agent, wi)
22.     end loop
23. end TicketAgent
```

Figure 4:TicketAgent Algorithm

The following are sequence of Telescript commands specification to notify an agent (for example, Bandwidth Agent) of target workstations.

Teleaddress addr := here@LookUpPlaces.Address.Copy();

Ticket(BandwidthAgent, addr):

SelfHomeAgent operates as follows:

a. Use SELF-command to activate autonomy.
b. Use GO command to activate the agent movement.
c. Use HOME command to activate return home.

LiveAgent uses the game of life algorithm presented in (Akinyokun, 1997) to account for the resource of the computer network environment.

MeetAgent operates using the command.

agentToMeet := here@MeetingPlace.Meet (aPetition, nil);

The agent will normally interact with the host operating system of the target and its appendages or utility programs for network monitor and cyber clock for the purpose of assessing and evaluating network resource. The results obtained by mobile agent after a successful visit to a set of target workstations are assembled using the specification of Figure 5 for the purpose of reporting them for external analysis, interpretation, policy formulation and decision making.

ASSEMBLE(SubAgent)

Agent ::= BandwidthEvaluator
AgentSet, AgentSet′ : ℙ Agent
SubAgent : array[1..n] of AgentSet
i : 1..n

∀ SubAgent : i → ℙ Agent
AgentSet′ ← AgentSet ∪ {SubAgent(i)}

Figure 5:Z Schema description of ASSEMBLE Agent

The disposal of the mobile agent simply means removing the mobile agent from the universal set of agents. The process of the mobile agent disposal is presented in Figure 6. Telescript command used to dispose agent is DUMP.

Agent

AgentName : seq CHAR

∀a, b : Agent • a ≠ b ⇒ a.AgentName ≠ b.AgentName

DisposeAgent

AgentSet, AgentSet′ : ℙ Agent

Remove? : Agent

Remove? ∈ Agentset
∃₁ k : Agentset • k.AgentName = Remove?.AgentName
AgentSet′ = AgentSet \ {Remove?}

Figure 6: Specification of the Mobile Agent Disposal Process

3 Conclusion

Formal method (Zed notations) has been used as the operational engine for demystification of the life cycle of a mobile agent. The use of Formal Method is one more weapon in the armoury against making design mistakes. It is hoped that developers of mobile agent system will find the scheme useful.

REFERENCES

[1]. Imianvan Anthony Agboizebeta and Akinyokun Oluwole Charles (2014),

[2]. Formal Characterization of a Mobile Agent Operational Environment, Journal of the Nigeria Association of Mathematical Physics, Published by Nigeria Association of Mathematical Physics, Volume 26, March 2014, pages 467 – 475

[3]. Imianvan Anthony Agboizebeta (2009), "Development of a Mobile Agent System for Evaluating the Use of Bandwidth in a Computer Network", PhD Thesis, Federal University of Technology, Akure, Ondo State. Nigeria.

[4]. Aderounmu G. A. (2001), "Development of an intelligent Mobile Agent for Computer Network Performance Management", PhD Thesis, Obafemi awolowo University, Ile-Ife, Nigeria.

[5]. Huy Hoang To, Shonali Krishnaswamy, and Bala Srinivasan (2005), Mobile Agents for Network Management: When and When Not! , ACM Syposium on Applied Computing.

[6]. Akinyokun O. C. (1997), "Catching and Using the Virus", The Journal of the Institute of the Management of Information Systems (IMIS), London, United Kingdom, Vol. 7, No. 6, Pages 12-17.

[7]. Weina He and Gaoyuan Liu (2011), The application of mobile agent in e-commerce, 3rd International Conference on Advanced Computer Control (ICACC), 2011, HARBIN.

[8]. Djamel Eddine Menacer, Habiba Drias, Christophe Sibertin-Blanc (2012), MP-IR: Market-Oriented Mobile Agents System for Distributed Information Retrieval, Advances in Intelligent and Soft Computing, Volume 122, pages 379-390, 2012.

[9]. Diller A., (1994), Z : An Introduction to Formal Methods, (2nd edition), John Wiley and Sons

[10]. Spivey J. M. (1998), "The Z notation: A Reference Manual", Prentice Hall International, United Kingdom.

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
VOLUME 3, ISSUE 2
ISSN 2054-7390

A Novel Approach to Compute Confusion Matrix for Classification of n-Class Attributes with Feature Selection

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

ABSTRACT

Confusion matrix is a useful tool to measure the performance of classifiers in their ability to classify multi-classed objects. Computation of classification accuracy for 2-classed attributes using confusion matrix is rather straightforward whereas it is quite cumbersome in case of multi-class attributes. In this work, we propose a novel approach to transform an n × n confusion matrix for n-class attributes to its equivalent 2 × 2 weighted average confusion matrix (WACM). The suitability of WACM has been shown for a classification problem using a web service data set. We have computed the accuracy of four classifiers, namely, Naïve Bayes (NB), Genetic Programming (GP), Instance Based Lazy Learner (IB1), and Decision Tree(J48) with and without feature selection. Next, WACM has been employed on the confusion matrix obtained after feature selection which further improves the classification accuracy.

Key words: Confusion Matrix, Classifiers, Feature Selection, Weighted Average Confusion Matrix, Classification Accuracy, Weighted average accuracy.

1 Introduction

Confusion matrix provides the basis for evaluating the performance of any classifier with the help of its four components, viz., True Positive (TP), False Negative (FN), False Positive (FP) and True Negative (TN). Among others, classification accuracy is the major parameter to judge the efficiency of a classifier. Classification accuracy of a classifier on a given data set refers to 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].

The common form of expressing classification accuracy is the error matrix (confusion matrix or contingency table). Error matrices compare the relationship between the known reference data and the corresponding results of classification on a class-by-class basis. The overall accuracy is computed by dividing the total number of correctly classified elements (the sum of the elements along the major diagonal) by the total number of elements in the confusion matrix. However, there are other contributing elements in the true negative component (which is an n-1 × n-1 matrix) of the confusion matrix which are ignored while computing the overall accuracy. This considerably reduces the accuracy of a classifier. The proposed WACM considers the contribution of those left out elements which eventually increases the accuracy of a classifier.

DOI: 10.14738/tmlai.32.1108
Publication Date: 28th April, 2015
URL: http://dx.doi.org/10.14738/tmlai.32.1108

In our earlier work [2], we have applied the weighted average technique for computing classification accuracy wherein the individual classification accuracy for each class of a multi-classed attribute is calculated first and then the individual accuracies of the respective classes are aggregated using the weighted average accuracy algorithm. This method has a limitation as it only helps in computing the classification accuracy. But, in order to calculate other performance criteria like sensitivity or true positive rate (TPR), specificity (SPC) or True Negative Rate, precision or positive predictive value (PPV), negative predictive value (NPV), fall-out or false positive rate (FPR), false discovery rate (FDR), Miss Rate or False Negative Rate (FNR), accuracy (ACC), F1 score, Matthews correlation coefficient (MCC), Informedness, Markedness etc. the four components, namely, TP, FN, FP and TN of a confusion matrix plays a vital role. In this work, we have proposed a technique to build a weighted average confusion matrix and have shown its novelty in the performance evaluation of four different classifiers, namely, Naïve Bayes (NB), Genetic Programming (GP), Instance Based Lazy Learner (IB1), and Decision Tree(J48). It is shown that the proposed approach considerably enhances the accuracy.

2 Weighted Average Confusion Matrix

2.1 Confusion Matrix

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 [3]. 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. In order that a classifier yields better accuracy it is necessary the total number of instances of a particular class in a data set should be represented along the diagonal of the confusion matrix (CM) as far as possible.

2.2 Confusion Matrix for two classes

Here we consider an example of a Two-class scenario that depicts the buying behavior of customers. Let us consider an attribute “buys computer” which can take two possible values “yes” and “no”. Next, we introduce the notion of positive tuples when the class attribute value is “yes” (i.e., buys computer = “yes”) and negative tuples when the class attribute value is “no” (e.g., buys computer = “no”). True positives refer to the positive tuples that were correctly labeled by the classifier as positive, while true negatives are the negative tuples that were actually labeled as negative 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”).

Table 1: Confusion matrix for 2-class scenario

		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 C1

False negative (FN) - The number of instances incorrectly classified as C2 (actually C1)

False positive (FP) - The number of instances incorrectly classified as C1 (actually C2)

True negative (TN) - The number of instances correctly classified as C2

P = Actual positive = TP + FN

P1 = Predicted positive =TP + FP

N = Actual negative = FP + TN

N1 = 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 / P1

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)

$$\text{F1 score} = \frac{2 \times precision \times recall}{precision + recall}$$

2.3 Conversion into 2 × 2 confusion matrix

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 [3]. 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: Confusion matrix for a three class scenario

		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. Considering the confusion matrix above, the corresponding table of confusion (Ref. Table 3), for the cat class, would be:

Table 3: Table of confusion for the class "Cat"

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 labelled as cats)	17 true negatives (all the remaining animals, correctly classified as non-cats)

Likewise, one can obtain 2 × 2 matrices for dog and rabbit classes. The actual count of a particular class is taken as the weight for the same class, e.g., for the "Cat" class the actual count is 8, hence the weight for the "Cat" class is taken as 8 at the time of building the WACM. Aggregating all the individual confusion matrices along with the weights of the individual classes, the weighted average confusion matrix for an attribute is calculated. The process of aggregation is presented in the following algorithm.

Weighted Average Confusion Matrix Algorithm:
Input: n × n Confusion Matrix
Output: Matrix containing elements of Weighted Average Confusion Matrix
WACM (A, CM)
//CM is n × n confusion matrix, where n is number of classes of an attribute, on which basis we //calculate classification accuracy. A is n+1 × n+5 matrix, where first n × 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,i) // A(i,n+2) is TP of individual class
For i=1 to n, A(i,n+3) = A(i,n+1) - A(i,n+2) // A(i,n+3) is FN of individual class
For i=1 to n, A(i,n+4) = A(n+1,i) - A(i,n+2) // A(i,n+4) is FP of individual class
For i=1 to n, A(i,n+5) = A(n+1,n+1) - $\sum_{j=n+2}^{n+4} A(i,j)$ //A(i,n+5) is TN of individual class
For i=n+2 to n+5, A(n+1,i) = $\sum_{j=1}^{4} [A(j,n+1) * A(j,i)]$ / A(n+1,n+1)
// A(n+1,n+2)….A(n+1,n+5) are 4 components(TP, FN, FP, TN) of 2 × 2 target confusion matrix
//Here A(j,n+1) is weight & $\sum_{j=n+2}^{n+5} A(n+1,j)$ is seen that it equals to A(n+1,n+1)
Return **A**
End

In the subsequent sections we show the applicability of WACM in improving the classification accuracy of classifiers. For our experimentation we have considered four different classifiers, namely, Naïve Bayes (NB), Genetic Programming (GP), Instance Based Lazy Learner (IB1), and Decision Tree(J48). First, we determine the classification accuracy of the individual classifiers and then apply feature selection to observe the improvement in accuracy. Finally, WACM is applied to compare the accuracy so obtained with the earlier experiments.

3 Classification Techniques

3.1 Naïve Bayesian Classifier

The Naïve Bayesian Classifier is one of the Bayesian Classifiers [4] which has been extensively used for classifying objects with a higher degree of accuracy. It has proven performance in various Machine Learning and Data Mining applications [5] - [8]. Naïve Bayesian classifiers assume that, given the class label all attributes within the same class are independent. Based on this assumption, the Naïve Bayesian classification rule is expressed as:

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

Where C represents a class label, Ai the attributes, and E the unclassified test instance. E is classified into class C with the maximum posterior probability.

3.2 Genetic Programming (GP)

It is a specialization of genetic algorithm (GA) [9]. Genetic Algorithm (GA) is a global search method based on natural selection procedure consisting of genetic operators such as selection, crossover and

mutation. GA optimizers are particularly effective in a high-dimension, multi-modal function, in which the number of variables tend to be higher. GA performs its searching process via population-to-population (instead of point-to-point) search. A member in a population called a chromosome is represented by a binary string comprising of 0 and 1 bits. Bits of the chromosome are randomly selected and the length of bit strings is defined in relevance. However, real values are taken in continuous genetic algorithm. In order to apply the methodology, a randomly generated initial population is required. From initial population, child population is born guided by three operators such as reproduction, crossover and mutation. New born child members are judged by their fitness function values. These child members act as parents in the next iteration. This procedure is repeated till the termination criteria are met.

The pseudo code of a genetic algorithm is depicted as below.

```
Simple Genetic Algorithm ( )
{
        Initialize the Population;
        Calculate Fitness function;
        While (Fitness Value! = Optimal Value)
        {
                Selection;
                Crossover;
                Mutation;
                Calculate fitness Function;
        }
}
```

3.3 Lazy Learner (IB1)

IB1 is a nearest-neighbor classifier [10] which uses normalized Euclidean distance to find the training instance closest to the given test instance, and predicts the same class as this training instance. If multiple instances have the same (smallest) distance to the test instance, the first one found is used.

The Euclidean distance between two points or tuples, say X1 = (x11, x12, …..x1n) and X2 = (x21, x22, …. x2n) is

$$\text{Dist}(X_1,X_2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2}$$

IB1 or IBL (Instance-Based Learning) is a comprehensive form of the Nearest Neighbor algorithm [10, 11]. IB1 generates classification predictions using only specific instances. Unlike nearest neighbor algorithm, IB1 normalizes the range of its attributes, processes instances incrementally and has a simple policy for tolerating missing values [10]. IB1 uses simple normalized Euclidean distance (similarity) function to yield graded matches between training instance and given test instance [11].

3.4 Decision Tree (J48)

J48 is a classifier for generating a pruned or unpruned C4.5 decision tree [12]. J48 is an open source Java implementation of the C4.5 algorithm [13] 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. The training

data is a set S = s1, s2, ….. of already classified samples. Each sample si consists of a p-dimensional vector (x1,i, x2,i, …, xp,i) where the xj represent attributes or features of the sample, as well as the class in which si falls.

4 Experimental Set Up

4.1 Data Set

The QWS (Quality of Web Service) dataset [14-16] consists of data from over 5000 web services out of which the public dataset consists of a random 364 web services. The service descriptions were collected using the Web Service Crawler Engine (WSCE) [17]. The majority of Web services were obtained from public sources on the Web including Universal Description, Discovery, and Integration (UDDI) registries, search engines, and service portals. The public dataset consists of 364 web services each with a set of nine Quality of Web Service (QWS) attributes that have been measured using commercial benchmark tools. Each service was tested over a ten-minute period for three consecutive days. WSRF is used to measure the quality ranking of a web service based on the nine quality parameters (1-9 in Table-4)

Table 4: QWS Parameter description

P-ID	Parameter Name	Description	Units
1	Response Time (RT)	Time taken to send a request and receive a response	ms
2	Availability (AV)	Number of successful invocations/total invocations	%
3	Throughput (TP)	Total Number of invocations for a given period of time	Invokes per second
4	Success ability (SA)	Number of responses / number of request messages	%
5	Reliability (REL)	Ratio of the number of error messages to total messages	%
6	Compliance (CP)	The extent to which a WSDL document follows WSDL specification	%
7	Best Practices (BP)	The extent to which a Web service follows WS-I Basic Profile	%
8	Latency (LT)	Time taken for the server to process a given request	ms
9	Documentation (DOC)	Measure of documentation (i.e. description tags) in WSDL	%
10	WSRF	Web Service Relevancy Function: a rank for Web Service Quality	%
11	Service Classification	Levels representing service offering qualities (1 through 4)	Classifier
12	Service Name	Name of the Web service	None
13	WSDL Address	Location of the Web Service Definition Language (WSDL) file on the Web	None

In table 4, the service parameters 1-9 are used for computation of classification accuracy with respect to four “Service Classification” values, namely, “Platinum” (high quality), “Gold”, “Silver” and “Bronze” (low quality) equivalent to 1 through 4 respectively. Thus, a classifier can give rise to a 4×4 confusion matrix.

4.2 WEKA Workbench

We have used the WEKA (Waikato Environment for Knowledge Analysis) machine learning platform [18] for our experimentation. The WEKA workbench consists of a collection of implemented popular learning schemes that can be used for practical data mining and machine learning.

4.3 Cross-Validation

We employ the cross-validation technique to calculate the accuracy of the classifiers. 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 wherein the dataset is partitioned into 10 subsets, of which 9 subsets are used as the 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 the mean of the estimates for each of the classifiers.

4.4 Feature Selection

An attribute selection measure is a heuristic for selecting relevant attributes and reducing redundant and irrelevant attributes in the dataset to improve upon classification accuracy. Therefore, suitable attribute selection method for selecting the most prominent features from the dataset is of paramount importance to enhance the performance of classification accuracy and reduce the computation time. In this study, we have applied two feature selection techniques, namely, Information Gain Attribute Evaluator and Gain Ratio Attribute Evaluator.

4.4.1 Information Gain

It evaluates the worth of an attribute by measuring the information gain with respect to a class. Information gain measure is used to determine how accurately a particular attribute classifies the training data. Information gain is based on the concept of entropy which is widely used in the Information theory domain.

Let node N represents the tuples of partition D. The attribute with the highest information gain is chosen as the splitting attribute for node N. This attribute minimizes the information needed to classify tuples in the resulting partitions and reflects the least randomness or impurity in these partitions [1].

The expected information needed to classify a tuple in D is given by

$$\text{Info (D)} = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

where pi is the probability that an arbitrary tuple in D belongs to class Ci and is estimated by |Ci,D| / |D|. Info(D) is the average amount of information needed to identify the class label of a tuple in D.

$$\text{Info}_A\text{ (D)} = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \text{Info (D}_j\text{)}$$

The term (|D_j |)/(|D|) acts as the weight of the j-th partition. InfoA (D) is the expected information required to classify a tuple from D based on the partitioning by A. Information gain is defined as the difference between the original information requirement and new information requirement. That is

$$\text{Gain (A)} = \text{Info(D)} - \text{InfoA(D)}$$

Using Information Gain Evaluation with Ranker Search method for web service data, top 3 attributes (WSRF, WSDL Address, Service Name) are selected for classification.

4.4.2 Gain Ratio

It evaluates the worth of an attribute by measuring the gain ratio with respect to the class.It applies a kind of normalization to information gain using a “split information” value. The split information value

represents the potential information generated by splitting the training data set D into v partitions corresponding to v outcomes on attribute A, and is expressed as [1]:

$$\text{SplitInfo}_A(D) = -\sum_{j=1}^{v} \frac{|D_j|}{|D|} \times \log_2\left(\frac{|D_j|}{|D|}\right)$$

The gain ratio is defined as

$$\text{GainRatio(A)} = \frac{\text{Gain(A)}}{\text{SplitInfo(A)}}$$

The attribute with the maximum gain ratio is selected as the splitting attribute.

Using Gain Ratio Evaluation with Ranker Search method for web service data, top 3 attributes (WSRF, Throughput, Response Time) are selected for classification.

5 Implementation of WACM Algorithm

For the chosen attribute "Service Classification" in the Web services data set the confusion matrix is a 4×4 matrix as the attribute can assume 4 possible class values, namely, 'Platinum', 'Gold', 'Silver' and 'Bronze'. Table-5, 6, 7 and 8 depict the 4×4 confusion matrix obtained using Naïve Bayes classifier for the attribute "Service Classification".

The intersection of 1st row and 1st column gives the TP value for the 'Platinum' class. Sum of the remaining elements in the 1st row gives the FN value and sum of the remaining elements in the 1st column gives the FP value for the 'Platinum' class. Similarly, sum of the remaining elements in the entire matrix gives the TN value for the 'Platinum' class. This is shown in table-5 and table-9(as described in table-3). Similarly, for 'Gold', 'Silver' and 'Bronze' class values the "2ndrow and 2nd column", "3rd row and 3rd column", "4th row and 4th column" are respectively consideration for determining the TP, FN, FP, and TN values. Instances are shown in table-6, table-7 and table-8 respectively.

Table 5: Instance for "Platinum"

41	0	0	0
1	94	5	0
0	1	119	0
0	0	0	103

Table 6: Instance for "Gold"

41	0	0	0
1	**94**	5	0
0	1	119	0
0	0	0	103

Table 7: Instance for "Silver"

41	0	0	0
1	94	5	0
0	1	**119**	0
0	0	0	103

Table 8: Instance for "Bronze"

41	0	0	0
1	94	5	0
0	1	119	0
0	0	0	**103**

Table-9 Table of confusion for the class "Platinum"

41 true positives (actual platinum that were correctly classified as platinum)	0 false negatives
1 false positive (gold that were incorrectly classified as platinum)	322 true negatives (all the remaining 'service classification' classes, correctly classified as non-platinum)

Next, we explain the process of obtaining the elements of Table 11.

The available 4 × 4 matrix is the actual data from the input 4 × 4 confusion matrix. First 4 elements of the 5th row represent the sum of column elements and the 5th column the sum of row elements. First 4 elements of the 6th column are for true positive values i.e. individual diagonal elements. First 4 elements of 7th column are for false negative values, which are obtained by subtracting TP from the concerned row sum. First 4 elements of 8th column are for false positive values, which can be obtained by subtracting TP from the concerned column sum. Lastly, first 4 elements of 9th column are for true negative values, which can be obtained by subtracting sum of the TP, FN, and FP from the total number of elements in the confusion matrix. In this way first 4 rows of columns 6, 7, 8 and 9 give TP, FN, FP and TN values for 'Platinum', 'Gold', 'Silver' and 'Bronze' classes respectively as shown in Table-10.

Table 10: Four 2 × 2 confusion matrices

Platinum		**Gold**	
41	**0**	**94**	**6**
1	**322**	**1**	**263**
Silver		**Bronze**	
119	**1**	**103**	**0**
5	**239**	**0**	**261**

Now to aggregate these individual confusion matrices to generate the resultant 2 × 2 confusion matrix, we have taken the actual number of instances for each class as the weight. As per the WACM algorithm, four components (TP, FN, FP and TN) of the WACM are calculated and placed in the last four cells of the last row of Table-11.

Table 11: (n+1) × (n+5) matrix as in algorithm

	Input confusion matrix				Row Sum	TP	FN	FP	TN
Platinum	41	0	0	0	41	41	0	1	322
Gold	1	94	5	0	100	94	6	1	263
Silver	0	1	119	0	120	119	1	5	239
Bronze	0	0	0	103	103	103	0	0	261
Column Sum	42	95	124	103	364	98.81868132	1.978021978	2.035714286	261.1675824

6 Results and Discussion

As explained in section 4, the four classifiers are first applied to classify the web services data set. Next, the same classifiers are used after applying two feature selection methods and the results are analyzed for possible improvement in the degree of accuracy. Finally, the process is repeated by applying the proposed WACM algorithm.

The sum of the 4 aggregated components TP, FN, FP and TN (the last row of table-11) turns out to be 364, which is the total number of instances in the data set. Classification accuracy i.e. (TP+TN) / (TP+FN+FP+TN) for the multi-classed attribute "Service Classification" is same as that calculated using the weighted average accuracy algorithm proposed in our earlier work [2] which establishes correctness of WACM.

Table 12: Classification accuracy for classifiers using IG

Classifier	Without feature selection	With feature selection (IG)	IG + WACM
NB	0.81044	0.980769	0.988973252
GP	0.983516	0.989011	0.993954535
IB1	0.717033	0.947802	0.969659461
J48	**0.997253**	**0.997253**	**0.998935817**

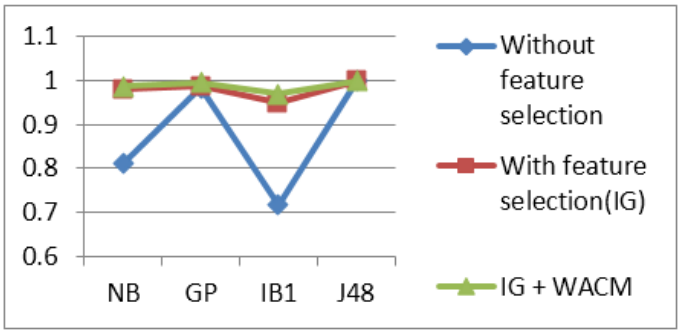


Figure 1: Improvement trend of classifiers

The values in Table-12 and 13 clearly indicate that by applying WACM the performance of the individual classifiers improves to a considerable extent irrespective of the feature selection method used. However, it is observed that both the feature selection methods do not have any impact on the J48 classifier, i.e., the classification accuracy remains unaltered with and without feature selection. Further, the classifier J48 outperforms all other classifiers in terms of accuracy. The improvement trends in both the cases are shown in figure-1 and 2.

Table 13: Classification accuracy of classifiers using GR

Classifier	Without feature selection	With feature selection (GR)	GR + WACM
NB	0.81044	0.887363	0.935824478
GP	0.983516	0.994505	0.997871634
IB1	0.717033	0.93956	0.96441402
J48	**0.997253**	**0.997253**	**0.998935817**

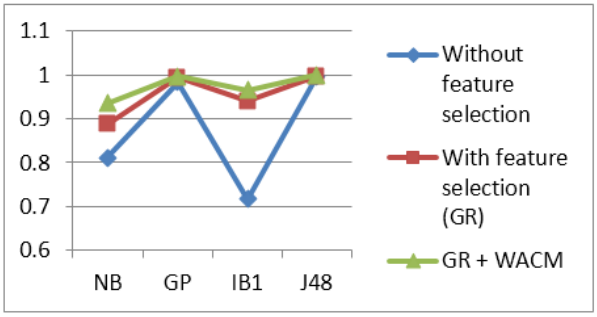


Figure 2: Improvement trend of classifiers

Further, the values for precision, recall and f-value are also computed using the weighted accuracy confusion matrix and are tabulated in table-14. Here also it is observed that the classifier J48 performs much better compared to the rest of the classifiers.

Table 14: Precision, Recall, F-value for the classifiers using WACM

Feature Selection	Classifier	Precison	Recall	F-Value
Information Gain	NB	0.979815314	0.980376124	0.980095639
	GP	0.990139576	0.988007632	0.989072455
	IB1	0.945970296	0.944371763	0.945170354
	J48	**0.998880729**	**0.997274462**	**0.998076949**
Gain Ration	NB	0.891105869	0.875197602	0.883080096
	GP	0.997757848	0.994548923	0.996150801
	IB1	0.935709808	0.935786318	0.935748062
	J48	**0.998880729**	**0.997274462**	**0.998076949**

7 Conclusion

In this work, we have introduced the notion of weighted average confusion matrix which is an aggregation of n number of 2 x 2 confusion matrices each referring to a particular class. Such a matrix enables one to compute the TP, FN, FP and TN components succinctly based on which the performance measures like Precision, Recall, F-value, etc. can be computed more accurately. In order to verify the usability of WACM, we have applied it for calculating the classification accuracy of four classifiers, namely, Naïve Bayes, Genetic Programming, Instance Based Lazy Learner, and Decision Tree. Feature selection techniques are also used to improve the accuracy. A systematic study shows that the performance of each of the classifier is improved to a considerable extent by using the weighted average confusion matrix. In future, we propose to study the impact of increasing the number of data instances on the accuracy level.

ACKNOWLEDGEMENTS

We would like to thank Dr. E. Al-Masri and Dr. Q.H. Mahmoud for providing us with the QWS dataset, which we have used for our experimentation.

REFERENCES

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

[2]. Patro, V. M., and Patra, M. R., 2014, "Augmenting Weighted Average with Confusion Matrix to Enhance Classification Accuracy", Transactions on Machine Learning and Artificial Intelligence, Volume 2 No 4, Aug (2014); pp: 77-91

[3]. http://en.wikipedia.org/wiki/Confusion_matrix last accessed on 10/11/14

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

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

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

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

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

[9]. Koza, J.R., 1992, "Genetic Programming: On the Programming of Computers by Means of Natural Selection", MIT Press.

[10]. Aha, D.W., Kibler, D., and Albert, M. K., 1991, "Instance-based learning algorithms", Machine Learning journal, Vol. 6, No 1, Page(s):37-66.

[11]. Witten, I. H., and Frank, E., 2005, Data Mining: Practical machine learning tools and techniques, 2nd Edition, Morgan Kaufmann, San Francisco, 2005.

[12]. Quinlan, J.R., 1993, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Francisco, CA.

[13]. http://en.wikipedia.org/wiki/C4.5_algorithm last accessed on 11/11/14

[14]. http://www.uoguelph.ca/~qmahmoud/qws/dataset/ last accessed on 04/09/14

[15]. Al-Masri, E., and Mahmoud, Q. H., 2007, "Discovering the best web service", (poster) 16th International Conference on World Wide Web (WWW), 2007, pp. 1257-1258.

[16]. Al-Masri, E., and Mahmoud, Q. H., 2007, "QoS-based Discovery and Ranking of Web Services", IEEE 16th International Conference on Computer Communications and Networks (ICCCN), 2007, pp. 529-534.

[17]. Al-Masri, E., and Mahmoud, Q.H., 2008, "Investigating Web Services on the World Wide Web", 17thInternational Conference on World Wide Web(WWW), Beijing, April 2008, pp. 795-804. (for QWS WSDLs Dataset Version 1.0)

[18]. www.cs.waikato.ac.nz/ml/weka/ last accessed on 14/11/14

SOCIETY FOR SCIENCE AND EDUCATION
UNITED KINGDOM
TMLAI TRANSACTIONS ON MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE
VOLUME 3, ISSUE 2
ISSN 2054-7390

Unified Acoustic Modeling using Deep Conditional Random Fields

Yasser Hifny
Department of computers and information systems, University of Helwan, Egypt;
yhifny@fci.helwan.edu.eg

ABSTRACT

Acoustic models based on Deep Neural Networks (DNNs) lead to significant improvement in the recognition accuracy. In these methods, Hid- den Markov Models (HMMs) state scores are computed using flexible discriminant DNNs. On the other hand, Conditional Random Fields (CRFs) are undirected graphical models that maintain the Markov properties of HMMs formulated using the maximum entropy (MaxEnt) principle. CRFs have limited ability to model spectral phenomena since they have single quadratic activation function per state. It is possible and natural to use DNNs to compute the state scores in CRFs. These acoustic models are known as Deep Conditional Random Fields (DCRFs). In this work, a variant of DCRFs is presented and connections with hybrid DNN/HMM systems are established. Under certain assumptions, both DCRFs and hybrid DNN/HMM systems can lead to exact same results for a phone recognition task. In addition, linear activation functions are used in the DCRFs output layer. Consequently, DCRFs and traditional DNN/HMM systems have the same decoding speed.

Keywords: Hidden Markov models; deep conditional random fields; deep neural networks; discriminative training.

1 Introduction

Acoustic modeling based on Hidden Markov Models (HMMs) [1, 2, 3, 4] is employed by state-of-the-art stochastic speech recognition systems. Generative HMMs are well understood models and may be trained efficiently using the Expectation-Maximization (EM) algorithm [5].

An example of an HMM with left-to-right transition topology, which is used to model a phone in an acoustic model, is shown in Fig. 1. This model has one entry state, three emitting states, and one exit state. The left-to-right topology imposes prior information, where speech production is sequential in time. For every observation at time *t*, a jump from the current state *i* to some new state *j* is allowed with a transition probability:

$$a_{ij} = P(\mathrm{s}_{t+1} = j | \mathrm{s}_t = i), \tag{1}$$

where, $\sum_j^N a_{ij} = 1, \mathrm{N}$ is the number of states in the HMM model. An acoustic feature vector O_t may be generated, with an output probability density function

DOI: 10.14738/tmlai.32.1124
Publication Date: 1st[th] May, 2015
URL: http://dx.doi.org/10.14738/tmlai.32.1124

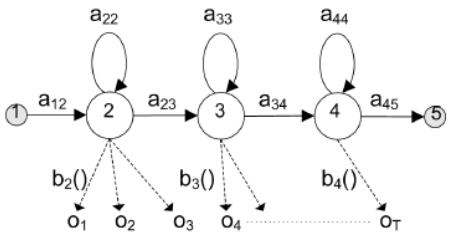


Figure 1: A typical Hidden Markov Model for a phone (a stochastic finite state machine view).

$b_j(o_t)$, which is associated with state j. A mixture of Gaussian distributions is typically used to model the output distribution for each state,

$$b_j(\mathbf{o}_t) = \sum_{m=1}^{M} c_{jm} \mathcal{N}(\mathbf{o}_t; \mu_{jm}, \Sigma_{jm}) \tag{2}$$

where M is the number of mixture components, c_{jm} is the component weight and $\sum_m^M c_{jm} = 1$. μ_{jm} and Σ_{jm} are the component specific mean vector and covariance matrix respectively. If the acoustic features are statistically independent, then diagonal covariance matrices are used to compute the likelihood of a Gaussian model,

$$\mathcal{N}(\mathbf{o}_t; \mu_{jm}, \Sigma_{jm}) = \prod_{d=1}^{D} \frac{1}{\sqrt{(2\pi)\sigma_{jmd}}} \exp\left(-\frac{(\mathbf{o}_{td} - \mu_{jmd})^2}{2\sigma_{jmd}^2}\right) \tag{3}$$

where σ_{jmd} is the variance element of the Gaussian component m for dimension d.

In hybrid ANN/HMM speech recognition systems [6], [7], arti_cial neural networks (ANN) models are used as exible discriminant classi_ers to estimate a scaled likelihood. In particular, the emission probability score is given by

$$b_j(\mathbf{o}_t) \approx \frac{P_\Lambda(\mathbf{s}_j|\mathbf{o}_t)}{P(\mathbf{s}_j)}, \tag{4}$$

where $b_j(o_t)$ is the score of state j in the traditional HMM framework, $P_\Lambda(\mathbf{s}_j|\mathbf{o}_t)$ is the posterior probability of a phonetic state estimated by a connectionist estimator [8],[9] and the prior $P(s_j)$ is estimated from the labeled data. In addition to discriminative training, if the posterior probability $P_\Lambda(\mathbf{s}_j|\mathbf{o}_t)$ is sensitive to acoustic context, $b_j(o_t)$ score may help to overcome conditional independence assumption and improve the overall recognition performance without changing the basic HMM framework. A graphical representation of the DNN/HMM acoustic model is shown in Fig. 2.

DNNs with several hidden layers that are trained using new methods have been shown to outperform Gaussian mixture models in several tasks [10], [11],

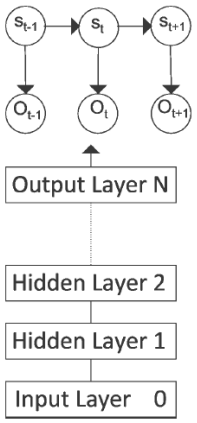


Figure 2: HMM model for phone representation, where the state scores are computed from a DNN.

[12], [13]. DNNs are trained in a generative way to learn the structure in the input data. This "pre-training" step provides a good initialization point to the traditional discriminative training using the backpropagation (BP) algorithm. DNN modeling is an active area of research and there is a lot of effort to improve the training speed of these models [14],[15], [16].

Over the last few years, there is an increased interest to develop acoustic models derived from MaxEnt [17, 18] and Conditional Random Fields [19]. Before CRFs became popular, there were several attempts to develop models similar to HMMs. In particular, the estimation of global posteriors using the forward-backward algorithm was derived in [20], [21]. Recent efforts in the field of MaxEnt/CRF modeling were reviewed and discussed in [22, 23]. Hidden Conditional Random Fields (HCRFs) were introduced to score the states based on a mixture of quadratic activation functions [24]. In [25], a multi-layer CRF model (deep-structured CRF) in which each higher layer's input observation sequence consists of the previous layer's observation sequence was presented. Deep extensions to HCRFs were developed in [26],[27]. A non-linear graphical model for structured prediction was introduced in [28]. In [29], deep hidden conditional neural fields (Deep-HCNF) which utilized an observation function with deep structure were presented. A segmental version of CRFs was developed in [30].

In [31, 32], a new acoustic modeling paradigm based on Augmented Conditional Random Fields (ACRFs) is investigated and developed. ACRFs paradigm addresses some limitations of HMMs while maintaining many of the aspects which have made them successful. In particular, the acoustic modeling problem is reformulated in a data driven, sparse, augmented space to increase discrimination. Acoustic context modeling is explicitly integrated to handle the sequential phenomena of the speech signal. In the context of ANN field, ACRFs can rep- resent CRFs with one hidden layer constructed from scoring a large number of Gaussians. Rank-based scoring used in maximum entropy direct modeling approaches [33, 34] may be interpreted as a mean to construct an augmented space.

Score-space kernels [35, 36], which are a generalization of the Fisher kernel [37], are used to extract new sufficient statistics, which may relax the conditional independence assumptions in a systematic fashion. These sufficient statistics are used to train MaxEnt models (C-Aug) for post-processing in HMM based speech recognition [38].

Training CRFs on the top of a hidden layer constructed from scoring a large number of sigmoid functions was introduced in [39]. One way to improve this approach is to compute the state scores based on a DNN that has several hidden layers. Hence, this improvement will lead to a deep version of CRFs

(DCRFs) [40]. In this work, a mathematical formulation of DCRFs is reviewed and connections with hybrid DNN/HMM systems are established. We will unify the training procedure between DCRFs and hybrid DNN/HMM in order to explore the gains related to different DNN structures used in the two systems. Under this assumption, the paper will show that the two systems can lead to same exact results for a phone recognition task. Consequently, DCRFs may be a natural choice for sequential modeling for speech recognition.

This paper is organized as follows: the basic limitations to use CRF as an acoustic model is addressed in Section 2. A mathematical formulation of DCRFs is described in Section 3. The discriminative training problem of DCRFs is addressed in Section 4. In Section 5, generative training which is used to initialize DNNs is presented. DCRFs and DNN/HMM systems compute the state scores using similar deep architectures. Hence, it is possible to unify and establish connections between DCRFs and DNN/HMM systems. This idea is addressed in Section 6. Section 7 gives experimental results on a phone recognition task. Several issues about the implementation of DCRFs are discussed in Section 8. Finally, a summary of the presented work is given in the conclusions.

2 Conditional Random Fields Limitations

Linear chain Conditional Random Fields are undirected graphical models that maintain the Markov properties of HMMs, formulated using the maximum entropy (MaxEnt) principle [41]. The maximum entropy formalism for sequential modeling results in a probability distribution, which is the log linear or exponential model:

$$P_\Lambda(\mathbf{S}|\mathbf{O}) = \frac{1}{Z_\Lambda(\mathbf{O})} \prod_{t=1}^{T} \exp\Big(\sum_j \lambda^j_{\mathbf{s}_t\mathbf{s}_{t-1}} a_j(\mathbf{s}_t,\mathbf{s}_{t-1}) + \sum_i \lambda^i_{\mathbf{s}_t} b_i(\mathbf{O},\mathbf{s}_t)\Big) \tag{5}$$

where

- $P_\Lambda(\mathbf{S}/\mathbf{O})$ obeys the *Markovian* property $P_\Lambda(\mathbf{s}_t|\{\mathbf{s}_\tau\}_{\tau\neq t},\mathbf{O}) = P_\Lambda(\mathbf{s}_t|\mathbf{s}_{t-1},\mathbf{O})$
- $\lambda^i_{S_t}$ *and* $\lambda^j_{S_tS_{t-1}}$ are the Lagrange multipliers (weighting factors) associated to the characterizing functions $b_i(O,s_t)$ and $a_j(S_t,S_{t-1})$.
- $Z \wedge (O)$ (Zustandsumme) is a normalization coefficient resulting from the natural constraints over the probabilities summation, commonly called the partition function and given by

$$Z_\Lambda(\mathbf{O}) = \sum_{\mathbf{S}} \prod_{t=1}^{T} \exp\Big(\sum_j \lambda^j_{\mathbf{s}_t\mathbf{s}_{t-1}} a_j(\mathbf{s}_t,\mathbf{s}_{t-1}) + \sum_i \lambda^i_{\mathbf{s}_t} b_i(\mathbf{O},\mathbf{s}_t)\Big)$$

and it is similar to the total probability $p(O|M)$ in HMMs, which can be calculated using the forward algorithm [19]. The conditional distribution behind the CRF model as shown in Equation (5) implies arbitrary combinations of state scores $b_i(O,s_t)$ and transition scores $a_j(s_t,s_{t-1})$. Hence, it is conceptually similar to HMMs that have only two scores; emission probability $p(o_t|s_t)$ and transition probabilities $P(s_t|s_{t-1})$ CRFs offer a principled framework for combining different state scores in a

natural way. The HMMs and CRFs share the first order Markov assumption, which simplifies the training and decoding algorithms.

CRFs have an attractive property: the MaxEnt models (linear chain CRFs are a special case) make little assumptions, as they are the most unbiased distributions that are simultaneously consistent with a set of constraints. Hence, CRF models do not suffer from the observation independence assumption made in the HMM framework, as the characterizing functions may be statistically dependent or correlated.

This is very clear in the model equation where the characterizing functions $b_i(O, s_t)$ are arbitrary functions over the entire observation sequence O. Moreover, CRF models do not constrain the shape of the data generation and the modeling quality is a function of the sufficient statistics represented by the characterizing functions. In speech recognition problems, second order sufficient statistics are extracted from the acoustic observations.

The state characterizing function $b_i(O, s_t)$ can depend only on the current observation (i.e. observation $b_i(O, s_t) = b_i(o_t, s_t)$). For example, frontend speech processing generally extracts MFCC+Δ+ Δ Δ as the basic acoustic vector, the observation dependent term in Equation (5) is given by

$$\begin{aligned}\sum_i \lambda^i_{\mathbf{s}_t} b_i(\mathbf{O}, \mathbf{s}_t) &= \sum_i \lambda^i_{\mathbf{s}_t} b_i(\mathbf{o}_t, \mathbf{s}_t) \\ &= \lambda^0_{\mathbf{s}_t} b_0 + \sum_{i=1}^{2d} \Big(\lambda^i_{\mathbf{s}_t} \mathbf{o}_{ti} + \lambda^i_{\mathbf{s}_t} \Delta \mathbf{o}_{ti} + \lambda^i_{\mathbf{s}_t} \Delta\Delta \mathbf{o}_{ti} \\ &\quad + \lambda^i_{\mathbf{s}_t} \mathbf{o}^2_{ti} + \lambda^i_{\mathbf{s}_t} \Delta \mathbf{o}^2_{ti} + \lambda^i_{\mathbf{s}_t} \Delta\Delta \mathbf{o}^2_{ti} \Big),\end{aligned} \tag{6}$$

where b_0 is the bias constraint, d is the vector dimensionality, and O_{ti}, O^2_{ti} are the first and second order moments of the acoustic features. Equation (6) can be written as

$$\sum_i \lambda^i_{\mathbf{s}_t} b_i(\mathbf{O}, \mathbf{s}_t) = \mathbf{o}^T_t \Lambda_{\mathbf{s}_t} \mathbf{o}_t + \lambda^T_{\mathbf{s}_t} \mathbf{o}_t + b_{\mathbf{s}_t 0}. \tag{7}$$

In addition, with one transition characterizing function, the transition dependent term in Equation (5) is given by

$$\sum_j \lambda^j_{\mathbf{s}_t \mathbf{s}_{t-1}} a_j(\mathbf{s}_t, \mathbf{s}_{t-1}) = \lambda_{\mathbf{s}_t \mathbf{s}_{t-1}} a(\mathbf{s}_t, \mathbf{s}_{t-1}) \tag{8}$$

where $a(s_t, s_{t-1})$ is a binary function and can be used to define CRF topology and $\lambda_{s_t s_{t-1}}$ is related to log $a_{s_t s_{t-1}}$ in HMM modeling. An example of a CRF with left-to-right transition topology, which is used to model a phone in an acoustic model, is shown in Fig. 3.

Equation (7) shows the main limitation of CRF model as used for speech recognition systems. This equation shows that state activation is based on a

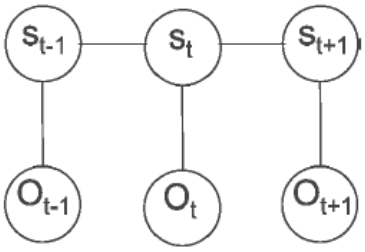


Figure 3: CRF model for phone representation

single quadratic activation function. In HMM context, this means the state score is based on a single Gaussian component. This low complexity model cannot model the spectral phenomena. Therefore, CRF acoustic models will lead to poor recognition results.

Hidden Conditional Random Fields (HCRFs) were introduced to score the states based on a mixture of quadratic activation functions [24]. This idea extends the CRFs to be similar to HMMs with mixture of Gaussians. However, the exponential quadratic activation functions are more flexible discriminant functions than Gaussian densities, which are used for local observation scoring within the HMM (but the physical meaning of mean and variance is no longer available). Alternatively, deep architectures can be used to compute the state scores. This idea is explored in the following section.

3 Deep Conditional Random Fields

Deep Conditional Random Fields acoustic models are a particular implementation of linear chain CRFs where the state scores are computed based on a DNN that has many hidden layers. The feed-forward phase updates the output value of each neuron. Starting from the first hidden layer, each neuron output is computed as a weighted sum of inputs and applying the sigmoid function to it:

$$\mathbf{o}_{tj}^{h} = \mathrm{sigm}(\sum_{i=1}^{n} \lambda_{ij} \mathbf{o}_{ti}^{h-1}) \tag{9}$$

where o_t^h is an output of a hidden layer, n is the number of inputs, h is an index to a hidden layer, and sigmoid function is computed as follows:

$$\mathrm{sigm}(x) = \frac{1}{1+e^{-x}} \tag{10}$$

The output of an hidden layer is passed to the next layer until the output layer is computed as follows:

$$\mathbf{o}_{tj}^{N} = \sum_{i=1}^{n} \lambda_{ij} \mathbf{o}_{ti}^{N-1} \tag{11}$$

where N is the index of the output layer. Hence, the activation of hidden layers is nonlinear based on a sigmoid function and the output layer activation is linear.

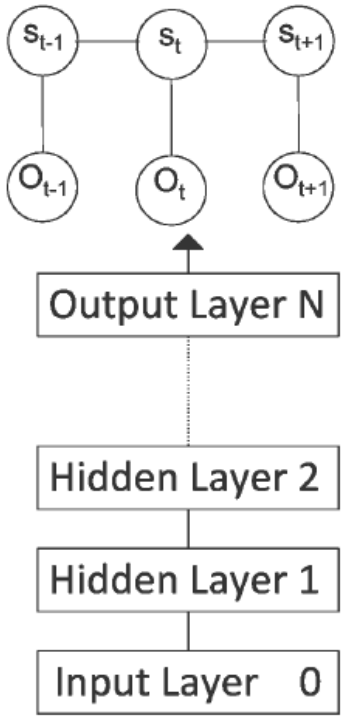


Figure 4: DCRF model for phone representation, where the state scores are computed from a DNN

A graphical representation of the DCRF acoustic model is shown in Figure 4. The conditional distribution defining DCRFs is given by

$$P_\Lambda(\mathbf{S}|\mathbf{O}) = \frac{1}{Z_\Lambda(\mathbf{O})} \prod_{t=1}^{T} \exp\left(\lambda_{\mathbf{s}_t \mathbf{s}_{t-1}} a(\mathbf{s}_t, \mathbf{s}_{t-1}) + b_{\mathbf{s}_t}(\mathbf{o}_t)\right) \tag{12}$$

where $b_s(o_t) = o_{ts_t}^N$ is computed from Equation (11). Hence, $b_{s_t}(o_t)$ connects DNN output to CRF input.

The partition function, $Z_\wedge(O)$, is given by

$$Z_\Lambda(\mathbf{O}) = \sum_{\mathbf{S}} \prod_{t=1}^{T} \exp\left(\lambda_{\mathbf{s}_t \mathbf{s}_{t-1}} a(\mathbf{s}_t, \mathbf{s}_{t-1}) + b_{\mathbf{s}_t}(\mathbf{o}_t)\right) \tag{13}$$

and it can be calculated using the forward algorithm [19].

4 DCRF Optimization

For R training observations {O_1,O_2,,,,,,,,,O_r,,,,,,O_R} with corresponding transcriptions {W_r}, DCRFs are trained using the conditional maximum likelihood (CML) criterion to maximize the posterior probability of the correct word sequence given the acoustic observations:

$$\begin{aligned} \mathcal{F}_{\mathrm{CML}}(\Lambda) &= \sum_{r=1}^{R} \log P_\Lambda(\mathcal{M}_{W_r}|\mathbf{O}_r) \\ &= \sum_{r=1}^{R} \log \frac{P(W_r) \sum_{\mathbf{S}|W_r} \exp \sum_t^T \Psi(\mathbf{O}, \mathbf{S}, c, \Lambda)}{\sum_{\hat{W}} P(\hat{W}) \sum_{\mathbf{S}|\hat{W}} \exp \sum_t^T \Psi(\mathbf{O}, \mathbf{S}, c, \Lambda)} \\ &\approx \sum_{r=1}^{R} \log Z_\Lambda(\mathbf{O}_r|\mathcal{M}^{\mathrm{num}}) - \log Z_\Lambda(\mathbf{O}_r|\mathcal{M}^{\mathrm{den}}), \end{aligned} \tag{14}$$

Where

$$\Psi(\mathbf{O},\mathbf{S},c,\Lambda)=\lambda_{\mathbf{s}_t\mathbf{s}_{t-1}}a(\mathbf{s}_t,\mathbf{s}_{t-1})+b_{\mathbf{s}_t}(\mathbf{o}_t) \quad (15)$$

The optimal parameters $\wedge^*$ are estimated by maximizing the CML criterion, which implies minimizing the cross entropy between the correct transcription model and the hypothesized recognition model. In other words, the process maximizes the partition function of the correct models[1] (the numerator term) $Z_{\wedge}(O_r \mid M^{num})$, and simultaneously minimizes the partition function of the recognition model (the denominator term) $Z_{\wedge}(O_r \mid M^{den})$. The optimal parameters are obtained when the gradient of the CML criterion is zero.

4.1 Numerical Optimization for DCRFs

Newton's method can be used to estimate DCRFs based on local quadratic approximation of the CML objective function. These methods rely on local quadratic approximation by expanding the CML nonlinear objective function $F_{CML}(\wedge+\delta)$ using Taylor expansion around the current model point $\wedge$ in parameter space and is given by

$$\mathcal{F}_{\mathrm{CML}}(\Lambda+\delta)\approx L(\Lambda)+\delta^T\mathbf{g}(\Lambda)+\frac{1}{2}\delta^T\mathbf{H}(\Lambda)\delta+\ldots, \quad (16)$$

where g($\wedge$) is the local gradient vector defined by

$$\mathbf{g}(\Lambda)=\frac{\partial\mathcal{F}_{\mathrm{CML}}(\Lambda)}{\partial\lambda_i}\Big|_{\Lambda} \quad (17)$$

and the H($\wedge$) is the local Hessian matrix defined by

$$\mathbf{H}_{ij}(\Lambda)\equiv\frac{\partial\mathcal{F}_{\mathrm{CML}}(\Lambda)}{\partial\lambda_i\partial\lambda_j}\Big|_{\Lambda} \quad (18)$$

The Newton's Method update rule is given by

$$\Upsilon_{(\perp)}=\Upsilon_{(\perp-\mathrm{I})}-\mathrm{I}_{(\perp)}\mathbf{H}_{-\mathrm{I}}(\forall)\mathfrak{R}(\forall) \quad (19)$$

Since CML is not a quadratic function, taking the full Newton step $H^{-1}(\wedge)g(\wedge)$ may lead to an overshoot of the maximum. Hence, $\eta^{(\tau)}\neq 1$ will lead to the damped Newton step. A line search algorithm is used to calculate $\eta^{(\tau)}$. A line search works by evaluating the objective function starting from the current model in the direction of search and choosing $\eta^{(\tau)}$ will lead to an increase of the CML objective function.

Hessian matrix calculation, its inverting and storage, makes Newton's Method useful only for small scale problems. Quasi-Newton or variable metric methods can be used when it is impractical to evaluate the Hessian matrix. Instead of obtaining an estimate of the Hessian matrix at a single point, these methods gradually build up an approximate Hessian matrix by using gradient information from some or all of the

[1] Since a summation over potential functions is commonly called the partition function in undirected graphical modeling, we coin the notation $Z_{\wedge}(O_r \mid M^{num})$ for the summation of all possible state sequences of the correct models.

previous iterates visited by the algorithm. Limited memory quasi-Newton's methods like L-BFGS are particular realizations of quasi-Newton's methods that cut down the storage for large problems [42].

Truncated-Newton method known as Hessian-Free approach [42, 43, 14], is a second order method for large scale problems. It finds the search direction using an iterative solver and the solver is typically based on conjugate gradient but other alternatives are possible. In this method, Hessian-vector products are computed without explicitly forming the Hessian. Hessian-free methods approximately invert the Hessian while quasi-Newton methods invert an approximate Hessian. By ignoring the second order derivative, a first order approximation of the CML will lead to the gradient ascent methods and the update is given by

$$\lambda^{(\tau)} = \lambda^{(\tau-1)} + \eta \mathbf{g}(\Lambda) \tag{20}$$

The step size η must be small enough to ensure a stable increase of the CML objective function. It can be shown that the algorithm is convergent provided that η satisfies the condition $0 \prec \eta \prec \frac{2}{\lambda_{\max}}$ where $\lambda_{\max}$ is the largest eigenvalue of the Hessian matrix $H(\wedge^{*})$ evaluated at the global maximum of the CML objective function [44]. In practice, second order statistics are not accumulated so $\lambda_{\max}$ is not known and η is chosen in an ad-hoc fashion by trial and error. The training speed of gradient descent (batch mode) is usually slow. The training process can be accelerated using an online variant known as stochastic gradient descent (SGD)[2]. This algorithm can update the learning system on the basis of the objective function measured for a single utterance or batch.

4.2 DCRFs Gradient Computation

For an exponential family activation function based on first-order sufficient statistics, the gradient of the CML objective function for the output layer parameters is given by

$$\nabla \mathcal{F}_{\mathrm{CML}}(\mathbf{O}) = \mathcal{C}_{ji}^{\mathrm{num}}(\mathbf{O}) - \mathcal{C}_{ji}^{\mathrm{den}}(\mathbf{O}) \tag{21}$$

where the accumulators of the sufficient statistics, $C_{ji}(O)$, for the j^{th} state and i^{th} constraint are calculated as follows:

$$\mathcal{C}_{ji}^{\mathrm{num}}(\mathbf{O}) = \sum_{r=1}^{R}\sum_{t=1}^{T_r} \gamma_j^r(t|\mathcal{M}^{\mathrm{num}})\mathbf{o}_{rti}^{\mathrm{N}} \tag{22}$$

$$\mathcal{C}_{ji}^{\mathrm{den}}(\mathbf{O}) = \sum_{r=1}^{R}\sum_{t=1}^{T_r} \gamma_j^r(t|\mathcal{M}^{\mathrm{den}})\mathbf{o}_{rti}^{\mathrm{N}}, \tag{23}$$

where ***r*** is the utterance index and the frame-state alignment probability γ^j, is the probability of being in state ***j*** at some time ***t*** can be written in terms of the forward score $\alpha_j(t)$ and the backward score $\beta_j(t)$ as in HMMs:

[2] Since CML objective function is maximized in this work, stochastic gradient ascent is used to train DCRFs models.

$$\gamma_j(t|\mathcal{M}) = P(\mathbf{s}_t = j|\mathbf{O};\mathcal{M}) = \frac{\alpha_j(t|\mathcal{M})\beta_j(t|\mathcal{M})}{Z_\Lambda(\mathbf{O}|\mathcal{M})}. \quad (24)$$

The delta of the output layer neuron ***j*** is given by

$$\delta_{tj}^N = \gamma_j(t|\mathcal{M}^{\text{num}}) - \gamma_j(t|\mathcal{M}^{\text{den}}) \quad (25)$$

and the delta of the hidden layers:

$$\delta_{tj}^h = \mathbf{o}_{tj}^h(1-\mathbf{o}_{tj}^h)\sum_{k\in\text{outputs}} \lambda_{kj}^{h+1}\delta_{kt}^{h+1}, \quad (26)$$

and the gradient for the hidden layers parameters is given by:

$$\frac{\partial \mathcal{F}_{\text{CML}}(\Lambda)}{\partial \lambda_{ki}^h} = \sum_{r=1}^{R}\sum_{t=1}^{T_r}\delta_{rtj}^h \mathbf{o}_{rtki}^{h-1} \quad (27)$$

Based on Equation (27) and Equation (21), a gradient based optimization can be used to estimate the parameters [42]. The transition parameters are given by:

$$\lambda_{\mathbf{s}_t\mathbf{s}_{t-1}} = \log a_{\mathbf{s}_t\mathbf{s}_{t-1}}, \quad (28)$$

where $a_{s_t s_{t-1}}$ is the transition probability in HMM modeling and is estimated using the maximum likelihood (MLE) criterion.

5 DCRFs generative training

The training of DNNs is divided into two phases: generative training to initialize the network to a good starting point, which may lead to good results. Fine tuning phase, which basically is the discriminative training described in Section 4. In this section, we will review the restricted Boltzmann machine (RBM), which is the basic building block for generative pretrained DNNs.

5.1 Restricted Boltzmann Machine

Restricted Boltzmann Machines (RBMs) are a special case of Markov random field that have one layer of binary stochastic hidden units and one layer of (Bernoulli or Gaussian) stochastic visible units. As shown in Fig. 5, they are bipartite graphs, where all visible units are connected to all hidden units. An RBM assigns an energy to every configuration of visible and hidden vectors, denoted v and h respectively according to

$$E(\mathrm{v},\mathrm{h};\theta) = -\mathrm{b}^T\mathrm{v} - \mathrm{c}^T\mathrm{h} - \mathrm{h}^T\mathrm{W}\mathrm{v}, \quad (29)$$

where W is the matrix of visible/hidden connection weights, b is the visible unit bias, and c is the hidden unit bias. The joint distribution $p(v,h;\theta)$ over the visible units v and hidden units h, given the model parameters θ, is defined in terms of an energy function $E(v,h;\theta)$ of

$$p(\mathrm{v},\mathrm{h};\theta) = \frac{\exp(-E(\mathrm{v},\mathrm{h};\theta))}{Z}, \quad (30)$$

where the partition function is given by

$$Z = \sum_{\mathrm{v},\mathrm{h}} \exp(-E(\mathrm{v},\mathrm{h};\theta)), \tag{31}$$

and the marginal probability that the model assigns to a visible vector v is

$$p(\mathrm{v};\theta) = \frac{\sum_{\mathrm{h}} \exp(-E(\mathrm{v},\mathrm{h};\theta))}{\sum_{\mathrm{h}} \sum_{\mathrm{u}} \exp(-E(\mathrm{u},\mathrm{h};\theta))}. \tag{32}$$

Since there is no hidden-hidden connections, the conditional distribution $p(h \mid v;\theta)$ is given by

$$p(\mathrm{v}=1|\mathrm{h};\theta) = \mathrm{sigm}(\mathrm{b} + \mathrm{h}^T \mathrm{W}^T). \tag{33}$$

Similarly, since there are no visible-visible connections, the conditional distribution $p(h \mid v;\theta)$ is given by

$$p(\mathrm{v}=1|\mathrm{h};\theta) = \mathrm{sigm}(\mathrm{b} + \mathrm{h}^T \mathrm{W}^T). \tag{34}$$

Although RBMs with the energy function of Equation (29) are suitable for binary input data, they cannot be used for real-valued input data. For example, frontend of a speech recognition system generates real-valued acoustic features. Therefore, the Gaussian- Bernoulli restricted Boltzmann machine (GRBMs) can be used to handle real-valued data. The GRBM energy function

$$E(\mathrm{v},\mathrm{h};\theta) = \frac{1}{2}(\mathrm{v}-\mathrm{b})^T(\mathrm{v}-\mathrm{b}) - \mathrm{c}^T\mathrm{h} - \mathrm{v}^T\mathrm{W}\mathrm{h}. \tag{35}$$

Note that Equation (35) implicitly assumes that the visible units have a diagonal covariance Gaussian noise model with variance 1 for each dimension. The corresponding conditional distributions are given by

$$p(\mathrm{h}=1|\mathrm{v};\theta) = \mathrm{sigm}(\mathrm{c} + \mathrm{v}^T\mathrm{W}), \tag{36}$$

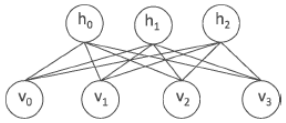


Figure 5: A graphical representation of Restricted Boltzmann Machine (RBM)

$$p(\mathrm{v}|\mathrm{h};\theta) = \mathcal{N}(\mathrm{v};\mathrm{b} + \mathrm{h}^T\mathrm{W}^T, I), \tag{37}$$

where ***I*** is the identity matrix. Apart from these differences, the inference and learning rules for a GRBM are the same as for a binary RBM.

5.2 RBM Training

Exact maximum likelihood learning of large RBM is not feasible because it is exponentially expensive to compute the gradient of the log likelihood of the training data. Instead, an efficient approximate training procedure called “contrastive divergence" (CD) can be used to train an RBM [45]. To compute the log likelihood, let us define a quantity known as the free energy:

$$F(\mathrm{v};\theta) = -\sum_{\mathrm{h}} \exp(-E(\mathrm{v},\mathrm{h};\theta)), \tag{38}$$

Using $F(v;\theta)$, we can write the log likelihood as:

$$L(\theta) = -F(\mathrm{v};\theta) - \log(\sum_{\nu} \exp(-F(\nu;\theta))). \tag{39}$$

Taking the gradient of the log likelihood $L(\theta)$ we can derive the update rule for the RBM weights as:

$$\frac{\partial L(\theta)}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \tag{40}$$

The first expectation $\langle v_i h_i \rangle_{data}$ is the frequency which the visible unit v_i and the hidden unit h_i are active together in the training data and $\langle v_i h_i \rangle_{model}$ is that same expectation under the distribution defined by the model. The one step CD approximation for the gradient w.r.t. the visible-hidden weights is:

$$\frac{\partial L(\theta)}{\partial w_{ij}} = \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} \tag{41}$$

where $\langle v_i h_i \rangle_1$ is the expectation over-one step reconstructions. In other words, it is the expectation computed with samples generated by running a Gibbs sampler initialized at the data for one full step. A Gibbs sampler can be defined using Equation (33) and Equation (34).

Once the gradient is computed, SGD can be used to update the RBM parameters. The update equation is given by

$$w_{ij}^{\tau} = w_{ij}^{\tau-1} + \alpha \frac{\partial L(\theta)}{\partial w_{ij}}. \tag{42}$$

6 Unified Frame Based Deep Acoustic Models

DCRFs and DNN/HMM systems compute the state scores using similar deep architectures. Hence, it is possible to establish connection between DCRFs and DNN/HMM systems. The difference between the two systems comes from three issues:

- The training criterion used to train each system.
- The state score of each system and this implies the output layer specifications of each system.
- The transition parameters of each system.

6.1 Training criterion

The traditional DCRFs are trained using sequence level CML training criterion to maximize the posterior probability of the correct word sequence given the acoustic observations as shown in Section 4. On the other hand, most DNN/HMM systems are trained using frame level CML training criterion (known as frame level cross entropy objective function)[3]. In order to get comparable results we need to unify the training criterion used to train the two systems. In this work, we will use the frame level CML training criterion to train the two systems. Therefore, the training of DCRFs needs to be modified to be based on frame level CML training criterion. Hence, the $\gamma_j(t \mid M)$ computation is approximated with state estimates as follows [47]:

[3] Hybrid DNN/HMM systems can be trained using a sequence training criterion [46].

$$\gamma_j(t|\mathcal{M}^{\text{den}}) = \frac{\exp\left(\mathbf{o}_{tj}^N\right)}{\sum_{\mathbf{s}} \exp\left(\mathbf{o}_{ts}^N\right)} \tag{43}$$

Based on this approximation, the training criterion used to train the two systems are identical and differences in the results related to the training criterion are eliminated.

6.2 DNN output layer

The conditional distribution defining hybrid DNN/HMM may be given by

$$P_\Lambda(\mathbf{S}|\mathbf{O}) = \frac{1}{Z_\Lambda(\mathbf{O})} \prod_{t=1}^{T} \exp\left(\lambda_{\mathbf{s}_t \mathbf{s}_{t-1}} a(\mathbf{s}_t, \mathbf{s}_{t-1}) + b_{\mathbf{s}_t}(\mathbf{o}_t)\right) \tag{44}$$

where $b_{s_t}(o_t) = \frac{P \wedge (s_t \mid o_{t)}}{P(s_t)}$. It is worth to mention that this conditional distribution is very similar to DCRF conditional distribution described in Equation (12). The only difference is how the state score is computed in each model. In hybrid DNN/HMM speech recognition systems, the HMM state scores are computed based on Equation (4). This equation implies the calculations

Table 1: Output layer design in different deep acoustic models.

System	Output layer score	Activation function
DCRF	$b_{\mathbf{s}_t}(\mathbf{o}_t) = \mathbf{o}_{ts_t}^N$	linear
DNN/HMM1	$b_{\mathbf{s}_t}(\mathbf{o}_t) = \frac{P_\Lambda(\mathbf{s}_t\|\mathbf{o}_t)}{P(\mathbf{s}_t)}$	softmax
DNN/HMM2	$b_{\mathbf{s}_t}(\mathbf{o}_t) = P_\Lambda(\mathbf{s}_t\|\mathbf{o}_t)$	softmax

of a softmax activation function for each frame to compute the state posteriors. On the other hand, DCRFs state scores are based on a linear activation function in the output layer based on Equation (11). Hence, it is possible to convert DNN/HMM systems to DCRFs by removing the output softmax layer and decode directly using the linear output activation. Due to the different output layer specifications, the DCRFs and hybrid DNN/HMM system may use different language scaling factor to lead to exact results. Therefore, in order to convert DNN/HMM system to DCRF system:

1. Train the DNN/HMM using frame cross entropy criterion.
2. Remove the softmax output layer.
3. Decode directly using the linear output activation.

Another form of DNN/HMM hybrid system is to assume that the state score is computed directly from the posterior probably of a connectionist estimator. In particular, $b_{s_t}(o_t)$ is given by:

$$b_{\mathbf{s}_t}(\mathbf{o}_t) = P_\Lambda(\mathbf{s}_t|\mathbf{o}_t). \tag{45}$$

This implies that $P(s_t)$ is a uniform distribution. It will be shown in the experimental section that these systems lead to exact results as DCRFs. Table 1 details the DNN output layer in the different systems under our unified deep acoustic modeling.

6.3 Transition parameters

The transition parameters may be a source of different results between DCRFs and DNN/HMM systems[4]. In order to unify the state scores between the two systems, the transition parameters should be identical for the two systems. This is easily achieved as described in section 4 by setting the transition parameters of the two systems using:

$$\lambda_{\mathbf{s}_t\mathbf{s}_{t-1}} = \log a_{\mathbf{s}_t\mathbf{s}_{t-1}}, \tag{46}$$

where $a_{s_t s_{t-1}}$ is the transition probability in HMM modeling and is estimated using the maximum likelihood criterion.

7 Experiments

We have carried out phone recognition experiments on the TIMIT corpus [48]. We used the 462 speaker training set, testing on the 24 speaker core test set, and the development set is based on 50 speakers from the test set [49]. The SA1 and SA2 utterances were not used. The speech was analyzed using a 25ms Hamming window with a 10 ms fixed frame rate. We represented the speech using 12 mel frequency cepstral coefficients (MFCCs), energy, along with their first and second temporal derivatives, resulting in a 39 element feature vector. Another representation is based on using a Log-Fourier-transform-based filter-bank with 40 coefficients (plus energy) distributed on a mel-scale, together with their first and second temporal derivatives resulting in a 123 element feature vector. The features are pre-processed to have zero mean and unit variance and acoustic context information is integrated using a window of 9 frames (4 left +current frame+ 4 right) to construct the final frames.

Following Lee and Hon [50], the original 61 phone classes in TIMIT were mapped to a set of 48 labels, which were used for training. This set of 48 phone classes was mapped down to a set of 39 classes [50], after decoding, and phone recognition results are reported on these classes, in terms of the phone error rate (PER), which is analogous to word error rate.

The baseline HMMs have three emitting states and the emission probabilities were modeled with mixtures of Gaussian densities with diagonal covariance matrices. The generative context-dependent HMMs (contained 1127 physical states, with 20 mixture components per state) were trained by the maximum likelihood criterion using the conventional EM algorithm [51]. The system is used only to provide the state alignment of the training data. Each phone was represented using a three state left-to-right DCRF, all parameters of DNN were initialized to random values and the transition parameters were initialized from trained HMM models forcing left to right DCRFs (the transition parameters are held fixed after the initialization). The training procedure accumulated the $\boldsymbol{M}^{num}$ sufficient statistics via a Viterbi pass (forced alignment) of the reference transcription using HMMs trained using maximum likelihood criterion. The language model scaling factor is set to 1.0 during the decoding process. All our experiments used a bigram language model over phones, estimated from the training set. In-house decoder is used to generate the recognition phone sequence.

[4] It is known that the transition scores have little impact on the recognition results.

For training DNNs, the PDNNTK toolkit is used [52] and it is based on Theano library [53], which supports transparent computation for CPUs and GPUs. In addition, the MFCC results is based on an in-house code developed based on Theano. In Table 2, DCRFs recognition performance is reported in terms of PER on TIMIT task (core test set) for MFCC based frontend.

The results based on filter bank frontend are shown in Table 3. Some DCRFs models were pretrained when the number of hidden layers is large. When the number oh hidden layers was 9, the LM scaling factor was set to 1:5.

It is possible to unify deep acoustic models based on a framework presented in section 6. However, the state score is different and may lead to different

Table 2: DCRF decoding results on TIMIT recognition task in terms of PER (MFCC based frontend).

#of Hidden layers	#of neuron	PER
1	8192	25.1%
2	3072	24.4%
3	3072	24.2%
4	3072	23.9%

Table 3: DCRF decoding results on TIMIT recognition task in terms of PER (FBANK based frontend).

#of Hidden layers	#of neuron	PER	Note
2	2048	24.2%	
4	3072	23.1%	
4	3072	23.0%	pretrained
5	3072	22.9%	pretrained
9	2048	**22.7%**	pretrained

decoding results. The DCRFs decoder is modified to support DNN/HMM1 and DNN/HMM2 decoding based on equations summarized in Table 1. As shown in Table 4, the decoding results are sensitive to the value of the language model scaling factor. It is clear that DCRFs and DNN/HMM2 hybrid systems lead to exact PER results.

8 Discussions

In this section we address several issues about the implementation of DCRFs.

8.1 Decoding speed

In hybrid ANN/HMM speech recognition systems, the HMM state scores are computed based on Equation (4). This equation implies the calculations of a softmax activation function for each frame to compute the state posteriors. However, in efficient implementations of DNN/HMM decoders, the softmax

Table 4: Comparison between different acoustic models using a unified frame-work on TIMIT recognition task in terms of PER (FBANK based frontend)

LM sacling factor	DCRFs	DNN/HMM1	DNN/HMM2
1	23.1%	24.3%	23.1%
1.5	22.7%	23.9%	22.7%
2	23.0%	23.8%	23.0%

calculations are ignored and the state scores are based on a linear activation function in the output layer. On the other hand, DCRFs state scores are based on a linear activation function. Consequently, DCRFs and traditional DNN/HMM systems have the same decoding speed.

8.2 Related prior work

The multilayer conditional random field (ML-CRF) was introduced in [39]. In this model, CRF is trained on the top of a single hidden layer constructed from scoring a large number of sigmoid functions. Hence, ML-CRF implies shallow neural networks. In addition, each phone was represented using a single state in the model. The Language model parameters are trained within the ML-CRF framework by defining bi-gram transition constraints. The training algorithm supports error backpropagation. In deep-structured CRF [25], multi-layer CRF model was developed where the marginal probabilities obtained from the outputs of a lower layer are used as the input of the higher layer. The model can be further extended for phonetic recognition using a variant called deep hidden conditional random field (DHCRF) [26]. In this model, the final layer is a Hidden Conditional Random Field (HCRF) [24] and the intermediate layers are zero-th-order CRFs. The DHCRF supports bi-gram language model features. Although the model has a deep architecture, it does not support DNN and the training algorithm does not support error backpropagation. DHCRFs were further modified to support DNN in [27], where state scores are computed based on DNN setup but the output layer has a softmax activation function. This version of the algorithm supports RBM training for initialization and error backpropagation training algorithm for fine tuning.

In [29], deep hidden conditional neural fields (Deep-HCNF) which utilized an observation function with deep structure were presented. The state scores are computed based on DNN setup and the output layer has a linear activation function as in [39] and our work. Deep-HCNF supports bi-gram language model features and Boosted-MMI training criterion (BMMI).

In this work, the state scores are also computed based DNN architecture and the output layer has a linear activation function. In addition, we do not estimate state transition parameters or language model parameters within DCRF frame-work. The state transition parameters were estimated using traditional HMM framework. Moreover, Maximum Likelihood (ML) criterion is used to estimate bigram language model. Hence, DCRF architecture may be computationally efficient for training and decoding. During the decoding process, a language model scaling factor is used to improve the results. On the other hand, frame level CML criterion is used to estimate DCRFs rather than the full-sequence CML training.

9 Conclusions

In this paper, we present a method to construct deep conditional random fields. In this approach, the state scores are computed based on a DNN that has many hidden layers. The feed-forward phase updates the output value of each neuron. Starting from the first hidden layer, each neuron output is computed as a weighted sum of inputs and applying the sigmoid function to it. The output is forwarded to the next layer until the output layer is updated as a weighted sum of inputs. DCRF state scores are connects the DNN output layer. Hence, the gradient is computed and a back-propagation algorithm is used to compute the gradient of each parameter in the hidden layers. It was shown in the paper , it is possible to unify the deep acoustic models under a variation of CRF framework. Under certain

assumptions presented in the paper, both DCRFs and hybrid DNN/HMM systems can lead to same exact results for a phone recognition task. In addition, linear activation functions are used in the DCRFs output layer. Consequently, DCRFs and traditional DNN/HMM systems have the same decoding speed. In addition, it is possible to convert DNN/HMM hybrid systems to DCRFs using a procedure addressed in the paper. On the other hand, we do not estimate state transition parameters or language model parameters within DCRF framework. The state transition parameters were estimated using traditional HMM framework. Moreover, Maximum Likelihood (ML) criterion is used to estimate bigram language model. Hence, the presented DCRF architecture may be computationally efficient for training and decoding.

REFERENCES

[1]. L. R. Rabiner, A tutorial on hidden Markov models and selected applications in speech recognition, Proc. of IEEE 77 (2) (1989) 257-286.

[2]. F. Jelinek, Statistical Methods for Speech Recognition, MIT Press, 1997.

[3]. X. Huang, A. Acero, H.-W. Hon, Spoken Language Processing: A Guide to Theory, Algorithm and System Development, Prentice Hall, 2001.

[4]. J. Bilmes, What HMMs can do, IEICE Transactions on Information and Systems E89-D (3) (2006) 869-891.

[5]. A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society 39 (1) (1977) 1-38.

[6]. S. Renals, N. Morgan, H. Bourlard, M. Cohen, H. Franco, Connectionist probability estimators in HMM speech recognition, IEEE Transactions on Speech and Audio Processing.

[7]. N. Morgan, H. Bourlard, Continuous speech recognition: An introduction to the hybrid HMM/connectionist approach, IEEE Signal Processing Magazine 12 (3) (1995) 25-42.

[8]. E. Trentin, M. Gori, A survey of hybrid ANN/HMM models for automatic speech recognition, Neurocomputing 37 (1-4) (2001) 91-126.

[9]. A. Robinson, An application of recurrent neural nets to phone probability estimation, IEEE Transactions on Neural Networks 5 (2) (1994) 298-305.

[10]. A. Mohamed, G. Dahl, G. Hinton, Acoustic modeling using Deep Belief Networks, IEEE Transactions on Audio, Speech and Language Processing 20 (2012) 14-22.

[11]. F. Seide, G. Li, D. Y. ., Conversational speech transcription using context-dependent Deep Neural Networks, in: Interspeech, 2011.

[12]. G. Dahl, D. Yu, L. Deng, A. Acero, Context-dependent pre-trained deep neural networks for large vocabulary speech recognition, IEEE Transactions on Audio, Speech, and Language Processing, Special Issue on Deep Learning for Speech and Language Processing.

[13]. G. Hinton, L. Deng, D. Yu, G. Dahl, A. rahman Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. Sainath, , B. Kingsbury, Deep Neural Networks for acoustic modeling in speech recognition, IEEE Signal Processing Magazine.

[14]. B. Kingsbury, T. N. Sainath, H. Soltau, Scalable minimum bayes risk train-ing of Deep Neural Network acoustic models using distributed hessian-free optimization, in: INTERSPEECH, 2012.

[15]. O. Vinyals, D. Povey, Krylov subspace descent for deep learning, in: AIS-TATS, 2012.

[16]. Y. Hifny, Deep learning using a Manhattan update rule, Deep Learning for Audio, Speech and Language Processing, ICML.

[17]. K. Van Horn, A maximum-entropy solution to the frame dependency problem in speech recognition, Tech. rep., Dept. of Computer Science, North Dakota State University (2001).

[18]. W. Macherey, H. Ney, A comparative study on maximum entropy and discriminative training for acoustic modeling in automatic speech recognition, in: Proc. EUROSPEECH, Geneva, Switzerland, 2003, pp. 493-496.

[19]. J. Lafferty, A. McCallum, F. Pereira, Conditional random _elds: Probabilistic models for segmenting and labeling sequence data, in: Proc. ICML, 2001, pp. 282-289.

[20]. J. Hennebert, C. Ris, H. Bourlard, S. Renals, N. Morgan, Estimation of global posteriors and forward-backward training of hybrid HMM/ANN systems, in: Proc. Eurospeech, Rhodes, 1997, pp. 1951-1954.

[21]. A. Krogh, S. K. Riis, Hidden neural networks, Neural Computation 11 (2) (1999) 541-563.

[22]. M. Gales, S.Watanabe, E. Fosler-Lussier, Structured discriminative models for speech recognition, IEEE Signal Processing Magazine.

[23].] E. Fosler-Lussier, Y. He, P. Jyothi, R. Prabhavalkar, Conditional random _elds in speech, audio, and language processing, Proceedings of the IEEE 101 (5) (2013) 1054-1075. doi:10.1109/JPROC.2013.2248112.

[24]. A. Gunawardana, M. Mahajan, A. Acero, J. Platt, Hidden conditional random fields for phone classification, in: Proc. INTERSPEECH, Lisbon, Portugal, 2005, pp. 1117-1120.

[25]. D. Yu, S. Wang, L. Deng, Sequential labeling using deep-structured conditional random fields, IEEE JOURNAL OF SELECTED TOPICS IN SIG-NAL PROCESSING.

[26]. D. Yu, L. Deng, Deep-structured hidden conditional random fields for phonetic recognition, in: Proc. INTERSPEECH, 2010.

[27]. A. Mohamed, D. Yu, L. Deng, Investigation of full-sequence training of Deep Belief Networks for speech recognition, in: Interspeech, 2010.

[28]. T.-M.-T. Do, T. Artieres, Neural conditional random fields, in: Proc. Of the 13th International Conference on Artificial Intelligence and Statistics,(AI-STATS), 2010.

[29]. Y. Fujii, K. Yamamoto, S. Nakagawa, Deep-hidden conditional neural fields for continuous phoneme speech recognition, in: Proc. IWSML, 2012.

[30]. G. Zweig, P. Nguyen, D. V. Compernolle, K. Demuynck, L. Atlas, P. Clark, G. Sell, M. Wang, F. Sha, H. Hermansky, D. Karakos, A. Jansen, S. Thomas, G. Sivaram, S. Bowman, J. Kao, Speech recognition with segmental conditional random fields: A summary of the JHU CLSP summer workshop, in: Proc. IEEE ICASSP, 2011.

[31]. Y. Hifny, Conditional random fields for continuous speech recognition, Ph.D. thesis, University Of Sheffield (2006).

[32]. Y. Hifny, S. Renals, Speech recognition using augmented conditional random fields, IEEE Transactions on Audio, Speech and Language Processing 17 (2) (2009) 354-365.

[33]. A. Likhododev, Y. Gao, Direct models for phoneme recognition, in: Proc. IEEE ICASSP, Vol. 1, Orlando, FL, USA, 2002, pp. 89-92.

[34]. J. K. Hong-Kwang, Y. Gao, Maximum entropy direct models for speech recognition, in: Proc IEEE ASRU Workshop, St. Thomas, U.S. Virgin Islands, 2003, pp. 1-6.

[35]. N. Smith, M. Gales, M. Niranjan, Data dependent kernels in SVM classification of speech patterns, Tech. Rep. CUED/F-INFENG/TR.387, University of Cambridge (2001).

[36]. N. Smith, M. Gales, Speech recognition using SVMs, in: Proc. NIPS, Vol. 14, 2002.

[37]. T. S. Jaakkola, D. Haussler, Exploiting generative models in discriminative classifiers, in: Proc. NIPS, Vol. 11, 1998.

[38]. M. Layton, M. Gales, Augmented statistical models for speech recognition, in: Proc. IEEE ICASSP, Vol. 1, France, 2006, pp. 129-132.

[39]. R. Prabhavalkar, E. Fosler-Lussier, Backpropagation training for multilayer conditional random _eld based phone recognition, in: Proc. IEEE ICASSP, Vol. 1, France, 2010, pp. 5534-5537.

[40]. Y. Hifny, Acoustic modeling based on deep conditional random fields, Deep Learning for Audio, Speech and Language Processing, ICML.

[41]. E. T. Jaynes, On the rationale of maximum-entropy methods, Proc. Of IEEE 70 (9) (1982) 939-952.

[42]. J. Nocedal, S. J. Wright, Numerical Optimization, Springer, 1999.

[43]. J. Martens, Deep learning via hessian-free optimization, in: Proc. ICML, 2010.

[44]. S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd Edition, Prentice Hal, 1998.

[45]. G. E. Hinton, Training products of experts by minimizing contrastive divergence, Neural Computation 14 (2002) 1771-1800.

[46]. B. Kingsbury, Lattice-based optimization of sequence classification criteria for neural-network acoustic modeling, in: Proc. IEEE ICASSP, 2009, pp.3761-3764. doi:10.1109/ICASSP.2009.4960445.

[47]. Y. Hifny, S. Renals, N. Lawrence, A hybrid MaxEnt/HMM based ASR system, in: Proc. INTERSPEECH, Lisbon, Portugal, 2005, pp. 3017-3020.

[48]. J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, V. Zue, TIMIT acoustic-phonetic continuous speech corpus (1990). URL http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC93S1

[49]. A. Halberstadt, J. Glass, Heterogeneous measurements and multiple clas-si_ers for speech recognition, in: Proc. ICSLP, Vol. 3, Sydney, Australia, 1998, pp. 995-998.

[50]. K.-F. Lee, H.-W. Hon, Speaker-independent phone recognition using hid-den Markov models, IEEE Transactions on Speech and Audio Processing 37 (11) (1989) 1641-1648.

[51]. S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, P. Woodland, The HTK Book, Version 3.1, 2001.

[52]. Y. Miao, PDNN: Yet Another Python Toolkit for Deep Neural Networks. URL http://www.cs.cmu.edu/ ymiao/pdnntk.html

[53]. J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Des-jardins, J. Turian, D. Warde-Farley, Y. Bengio, Theano: a CPU and GPU math expression compiler, in: Proceedings of the Python for Scientific Com-puting Conference (SciPy), 2010, oral Presentation.