Proceedings of the Ninth York Doctoral Symposium on Computer Science & Electronics

Department of Computer Science & Department of Electronics
The University of York
York, UK

17th November 2016

Editors: James Baxter and Samuel Bourke

Organisation

General Chairs

Matthew Dale
Richard Redpath

Organising Committee

Saja Althubaiti
James Baxter
Matthew Bedder
Samuel Bourke
Matthew Dale
Mark Douthwaite
Penny Faulkner
Fatma Layas
Matthew Prinold
Richard Redpath
Matthew Rowling
Martin Rusilowicz

Programme Chairs

James Baxter
Samuel Bourke

Programme Committee

Anil Bas
James Baxter
Matthew Bedder
Samuel Bourke
Matthew Dale
Amir Dehsarvi
Mark Douthwaite
Penelope Faulkner
Simos Gerasimou
Russell Joyce
Fatma Layas
Matt Luckcuck
Hashan Mendis
Nils Morozs
Stephen Oxnard
Sarah Parker
Colin Paterson
Richard Redpath
Matt Rowlings
James Stovold
Matt Windsor

Sponsoring Institutions

The Organising Committee are grateful to the following organisations for their generous sponsorship of YDS 2016. Without such sponsorship YDS would not be able to provide doctoral students of Computer Science and Electronics the opportunity to experience such a well rounded, fully edged academic conference in their field as a training experience.

Academic Sponsors

The Organising Committee would like to thank the departments of Computer Science and Electronics, and the Vice Chancellor's department for their support.

Industrial Sponsors

The Organising Committee would like to thank the following organisations for their sponsorship.


Sim Omics


STAR
COMPLIANCE


THALES

Preface

The York Doctoral Symposium on Computer Science and Electronics (YDS) is an international post-graduate student symposium, now in its ninth year. YDS 2016 was organised jointly by the departments of Computer Science and Electronics, as it has been for the past three years. The primary aims of the symposium are to provide an opportunity for early-stage researchers to observe and take part in the process of academic publication through organising a full-scale international academic event, and to allow doctoral students from the UK and Europe to experience a peer-reviewed conference where they can share and exchange their research and ideas in a supportive environment. Furthermore, by bringing together students from a wide range of areas, YDS also hopes to promote more interdisciplinary research.

YDS 2016 offered three categories of submissions: full-length papers, extended abstracts and posters. We received 17 full-length papers, 5 extended abstracts and 12 abstracts for poster submissions. We received more full-length paper submissions than any previous year. The submissions were well distributed between the areas of computer science and electronics. Furthermore, over half the total submissions were from external universities, including 3 full-length paper submissions from outside the UK. The acceptance rate for full-length papers and extended abstracts combined was 50%, including one paper that was accepted but later withdrawn as the author was unable to arrange travel to attend the conference. We offered the opportunity for some of the rejected submissions to be presented as a poster instead. It was important that submitted work was reviewed with due anonymity and fairness, as the event is a serious training exercise for the organisers as well as for the contributors and attendees. Therefore, all full-length papers and extended abstracts received three anonymous reviews each.

We are very grateful to our sponsors, without whom YDS could not take place. YDS 2016 was sponsored by the departments of Computer Science and Electronics, the Vice Chancellor's department, and our industrial sponsors: ETAS, IBM, SimOmics, Star Compliance, and Thales. Among other things, their financial help enabled us to offer prizes for the best full-length paper, extended abstract, poster, and presentation. We were honoured to host invited keynote talks by Prof. Steve Furber, from the University of Manchester, and Dr. Martin Trefzer, from the University of York, who kindly stepped in after one of our keynote speakers fell ill.

I would like to express my gratitude to all who served on the YDS 2016 organising committee, for their hard work in putting the event together, particularly the chairs Matt Dale and Richard Redpath, who worked tirelessly to make sure all the aspects of the conference were handled. I am also grateful to all the members of the programme committee for their time and effort in producing reviews. I am particularly pleased at how much students from the Department

of Electronics contributed to the organisation of YDS 2016. I would also like to thank all the staff who supported us in organising YDS 2016, particularly Dr. Mike Dodds, who was always friendly and willing to help, even when we had some difficult situations.

YDS 2016 has been an exhausting but exciting and educational endeavour. The chance to chair the programme committee of an academic conference is hugely beneficial and I am glad to have taken the opportunity. I would encourage all post-graduate students to participate in YDS, whether by organising, reviewing submissions, contributing submissions, or simply attending. Finally, I would like to wish the organising committee for YDS 2017 all the best for the next year.

James Baxter
YDS 2016 Programme Chair

Table of Contents

IV Poster Abstracts

Part I

Keynote Talks

Engineering Taught by Nature: Biologically Inspired Electronic Systems

Martin Trefzer

University of York

Abstract. The increasing versatility, performance, compactness and power efficiency of todays electronic systems is achieved by pushing technology to its physical limits: systems are increasing in size and complexity comprising thousands of subsystems made of billions of devices, requiring sophisticated programming and control; the devices themselves become smaller and smaller and have reached the atomic scale, which leads to stochastic variations when fabricating them. This makes components more noisy and unreliable and designing reliable systems extremely challenging. In this respect, technological systems are far behind biological organisms which have long since accomplished the feat of not only operating reliably with highly variable components, but also maintaining and tuning themselves in changing environments, when faults occur or they are otherwise perturbed. Biological mechanisms enabling this have co-evolved with the organisms, hence, are perfectly adapted to the requirements of their embodiment. In this context, evolutionary hardware is about hardware that offers the capability to change its structure and behaviour in order to automatically optimise its operation for a specific task or environment, taking inspiration from biological organisms with natural evolution as Natures guiding optimisation principle. The talk will give examples of hardware systems, biological systems, and how the former can learn from the latter.

Building Brains

Steve Furber

University of Manchester

Abstract. The inner workings of the human brain remain a scientific enigma, but there is a growing global consensus that the time is right to employ the formidable computer power now available to us to try to unlock some of the brains secrets. The SpiNNaker (Spiking Neural Network Architecture) project aims to produce a massively-parallel computer capable of modelling large-scale neural networks in biological real time. The machine has been 20 years in conception and ten years in construction, and has far delivered a 500,000-core machine in five 19-inch rack cabinets, which is now being expanded towards the million-core full system. In this talk I will present an overview of the machine and the design principles that went into its development, and I will indicate the sort of applications for which it is proving useful.

Part II

Full-length Papers

Gaining Insight from Student's Use of Source Control

Simon Grey

University of Hull, Hull, UK,
S.Grey@hull.ac.uk

Abstract. For programmers experience of the use of software source control is a valuable professional skill and so it is vital that computer science students are exposed to source control systems before the graduate in order to enhance their employability. Additionally, there are additional benefits of students using source control for both students and teachers. For students source control provides a low risk environment and allows experimentation. For teachers source control systems can be used as a convenient medium to deliver scaffolding code, and allows the teacher access to student code in order to provide assistance to student's who need it. Additionally each student's use of source control provides valuable information about that student's behaviour and engagement with the work. This paper will begin by presenting a literature review of previous research use of source control systems by teachers and students. Following that some preliminary analysis of empirical data collected from student's interaction with subversion will be presented.

Keywords: source control, subversion, SVN, analytics, learning, teaching

1 Introduction

Experience of using source control and an appreciation of the benefits source control provides are valuable professional skills for computer science graduates. As well as its intended purpose of facilitating group work use of source control has significant potential benefits for individuals who are just learning to program. Anecdotal evidence shows that students who are learning basic programming concepts can shy away from experimentation because they are fearful that doing so may break their code. Source control can also provide students who are learning to program with a low risk environment in which to experiment freely - a vital part of the learning process - safe in the knowledge that their is a quick and easy way to undo any undesirable changes.

The remainder of this paper is split into three sections. Immediately following this one, in section 2 a literature review of publications concerning the use of teachers use of source control with their students. Following that, in section 3 a case study of use of source control on a second year graphics and simulation module is presented, together with some preliminary results. Finally, in section 4 conclusions and further work will be discussed.

2 Background

For computer science students an appreciation of the intricacies of source control is a valuable professional skill. Furthermore, in a learning environment source control can be used for more than just source code management.

This section provides an overview of published research concerning the use of source control management tools by students. In the next section 2.1 an overview over the various source control management tools will be presented.

2.1 Source Control Management Tools

The primary purpose of source control management tools is to enable multiple software developers to work on the same code base at the same time. As well as editing source code they also offer functionality to enable developers to get the latest changes, undo changes or merge changes to files that have been edited by two developers simultaneously. They serve as a record through time of the state of the source code.

Although Koc and Tansel identify four models for version control [11] two of those models are most prevalent. They are the *client/server* or *centralized* model and the *distributed* model. The centralized version control systems (CVCS) such as Subversion and CVS. Git and Mercurial are examples of distributed version control systems (DVCS).

The CVCS model uses the concept of a centralized, *golden copy* of the repository that exists on a central server. Individual developers *check out* a *working copy* of the golden copy. When changes are made they are integrated with the central golden copy via a *commit* command. Changes can be pulled from the golden copy down to the working copy by performing an *update*.

In the DVCS model each developer *branches* a local copy of the repository. Commits are made locally, and later the changes can be *pushed* back to the original branch. In recent years there has been a move away from the CVCS model and towards the DVCS model [1, 11] with Brindescu et al reporting that in a survey of 820 developers 65% use DVCS and 35% CVCS, and that the most popular source control solutions were Git (DVCS, 52%) and SVN(CVCS, 20%)[1].

2.2 Usage of Source Code Management Tools

Source control management tools have been used in a variety of ways in education, beyond the original purpose of facilitating group work. Of course, using source control to help facilitate student group work is still a valuable exercise. Beginning with group work this section presents a literature review of ways in which teachers have used source control in their teaching.

Group Work

Helping to facilitate group work is probably the most natural usage of source control. Using source control in this way is likely to give students a more authentic experience of source control that will arguably result in more employable

students. Much of the research concerning student's use of source control centres around group work [2, 3, 7, 12–14, 17].

2.3 Individual Work

Even though source control has been designed with groups of developers in mind, using source control also has benefits for student's working as individuals. The body of research into student's use of source control also includes a good amount of student's using their own individual repositories [4–6, 9, 16, 18, 19]. Storing work in a remote repository, rather than on the student's own hardware, or a thumbstick means that there will likely be a robust back up process in place. This protects students from loss of work through hardware failure. Additionally, and perhaps more importantly for students who are just learning how to write code use of source control allows freedom of experimentation. Anecdotal evidence tells stories of students who have broken working code, and have had to spend a lot of time trying to get back to a functional state. Not only do these students view this time as wasted, but they are also reluctant to make further changes or experiment with their code for fear of breaking it again.

2.4 Delivery of Material

Source control can also offer a more appropriate mechanism for delivery of course material to computer science students that is usually available in a VLE. Clifton et al give a comprehensive account of the benefits of using Subversion as part of course management[2]. When working with code based solutions spread across multiple files there seems to be pedagogic value in providing a good deal of scaffolding in order to remove barriers such as downloading, unzipping, linking and compiling a solution that separate a student from the practical application of learning outcomes they should be considering. This can be achieved by committing solutions into individual student repositories. It should be noted, however, that this treatment robs students of an opportunity to work with source control more deeply. Glassy required students to perform additional tasks such creating their repositories and importing code into them citing that students gained extra benefits because *"it forces students to become acquainted with all stages of version control usage."*[4].

2.5 Delivery of Assistance

Kertész describes, among others, the benefits of using source control to enable fast formative feedback from both student peers and instructors [9, **?**]. This is supported by anecdotal evidence suggests that there is an added benefit of source control for students seeking assistance from instructors. If a student experiences a programming problem they are able to get help either by email or in person. Students effective use of source control means that instructors also have access to that source code and are better able to determine exactly what problem the student is facing and provide guidance as is deemed appropriate.

2.6 Submission of Assessments

An extremely popular use of source control is to enable students to submit their assessed work for marking [2, 3, 5–8, 10, 12, 14, 15, 17–19]. In some cases subversion was also used to deliver formal feedback to students [2]. Additionally Gregorio and González-Barahona discus to potential for integrating plagiarism detection tools into student's software repositories.

2.7 Monitoring and Analytics

Often source control management systems include comprehensive logs of activity affecting the repositories of source code. This generates huge amounts of data for analysis. Monitoring students use of source control and analysing the data generated has been a focus of many studies [3, 4, 7, 9, 10, 13–16, 18]. Ganapathy et al [3] and Kim et al [10] present visualisation of student activity to instructors as a way of monitoring activity, however instructors commented that whilst this information was useful it was sometimes misleading - the number of commits was not a reliable proxy for student effort but sometimes a lack of commits was cause for concern. Jones[7] describes a simple process of performing a weekly manual review of student activity in groups using existing built in monitoring tools for Subversion. Novak et al[18] present a system for tracking student's interactions with Git based source control and visualize them using GitLab.

Mierle et al[16] mined student CVS repositories looking for performance indicators. They were unable to use metrics gathered from student source control to accurately predict performance, but did find a correlation between student performance and lines-of-code written. This result will be reaffirmed later in sections 3.2. Ljubovic and Nosovic[15] report a correlation between lines of code as reported by analysis software and time students reported they spent working on that code.

2.8 Summary

To summarize the key choices when choosing a source control solution have been outline in subsection 2.1 mostly centring around a decision between centralised and distributed source control. In subsection 2.2 a variety of ways to use source control are discussed with reference to literature. Aside from the main functions of managing source code and facilitating group work key usages include delivery of course materials and assistance, submission of assessment and monitoring student engagement. In the next section 3 a case study will be presented including preliminary analysis of data from three student cohorts' interaction with source control.

3 Case Study

This section will present data collected from a second year module in Simulation and 3D Graphics collected from three separate cohorts during 2014, 2015

and 2016. Subsection 3.1 includes background information concerning the setup of source control system and how it was used, as well as the tool chain used to extract data. Following that in subsection 3.2 some preliminary results are presented and analysed.

3.1 Setup

At the University of Hull students are provided with Subversion repositories despite it not being the most popular solution, or even version control model. Several factors were considered when choosing an introductory level VCS[1]. Polling student opinion gave no clear consensus. However as a centralized solution Subversion offers a clearer picture of student engagement through interaction with the repository. The DVCS model would allow students to gain all the advantages of using a VCS and making local commits without pushing changes to the server. This may also put the student's work at risk if they do not back up their local copy with the same regularity as the centralised repositories are backed up. Finally, it is perceived that Subversion is easier to understand than a DVCS, especially in the context of an individual developer rather than a group. Using a product called VisualSVN [2] allows adminstration and integration of student repositories using existing Microsoft based services including student authentication through Active Directory. Although we chose to host and administer our own source control solution it is worth acknowledging there are advantages to outsourcing this work. Lawrance et al [12] in particular describe the ease of using GitHub to provide repositories for education.

For each cohort a repository was created for the whole module. Within this shared repository students each have their own folder. Students have read and write permissions to their own folder, but cannot read or write to folders belonging to other students. Scaffolding code for laboratory assignments is committed to every students folder using a script. Laboratory assignments are guided and prompt students at to commit at appropriate times. Course work submission is also performed through the same repository. Additionally, within each student folder there was a read only folder that was used for delivery of feedback.

The laboratory work and assessment students complete is the same each year and so the years are directly comparable. As a measure against students copying each others work each student is given an individual data driven specification for their assessment created using their student ID as a random seed. This is commited to the read only folder in their repositories.

3.2 Results and Analysis

Data is extracted from the repositories using the StatSVN tool [3]. StatSVN is one of the tools suggested by Ljubovic et al [15]. StatSVN extracted data from

[1] Students are exposed to Team Foundation Server in later year as a project lifecycle management tool providing more than just version control.
[2] https://www.visualsvn.com/
[3] http://statsvn.org/

CSharp and shader code within the directory structure and ignored any output directories. Without explicitly ignoring these directories the results are skewed by students who committed intermediate files generated by the compilation process to the repository. Initially StatSVN was used to collect the Lines of Code changed by each student. It is worth noting that this metric is cumulative over all commits. That is, if a student were to change the *same line* ten times they would record 10 lines of code changed.

Students who failed to engage with assessment were removed as they have no mark. Students who interacted with source control outside of the normal semester, for example because they did a resit or a fresh attempt were also removed. A total of 185 students were remaining who between them made 11,605 commits changing 561,000 lines of code.

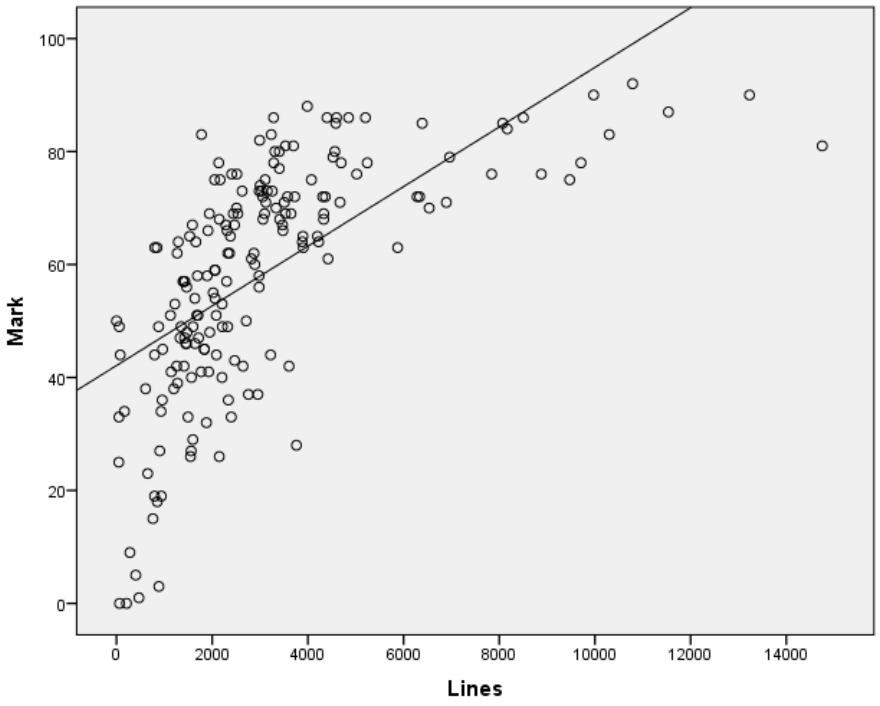


Fig. 1. Lines of Code Changed versus Mark Achieved

Figure 1 shows lines of code changed against mark achieved. A two tailed pearson correlation shows a strong positive correlation between lines of code and marks achieved of 0.64. The significance of the correlation was very significant at the 0.01 level. A linear line of best fit has been added to the graph. Although a cubic line provides a better fit, it is likely that this is due to the nature of the

course work, which is designed such that there is a greater weight towards easier functionality to ensure that all students who engage should be able to achieve something. Functionality that is more difficult to achieve carries fewer marks. This functionality is intended to challenge the more advanced students.

There appear to be a few outliers from students who wrote made far more changes that other students. Twenty students edited more than 6,000 lines of code over the course of the module. If these students are removed the pearson correlation increases to 0.70. The graph of this data is shown in figure 2.

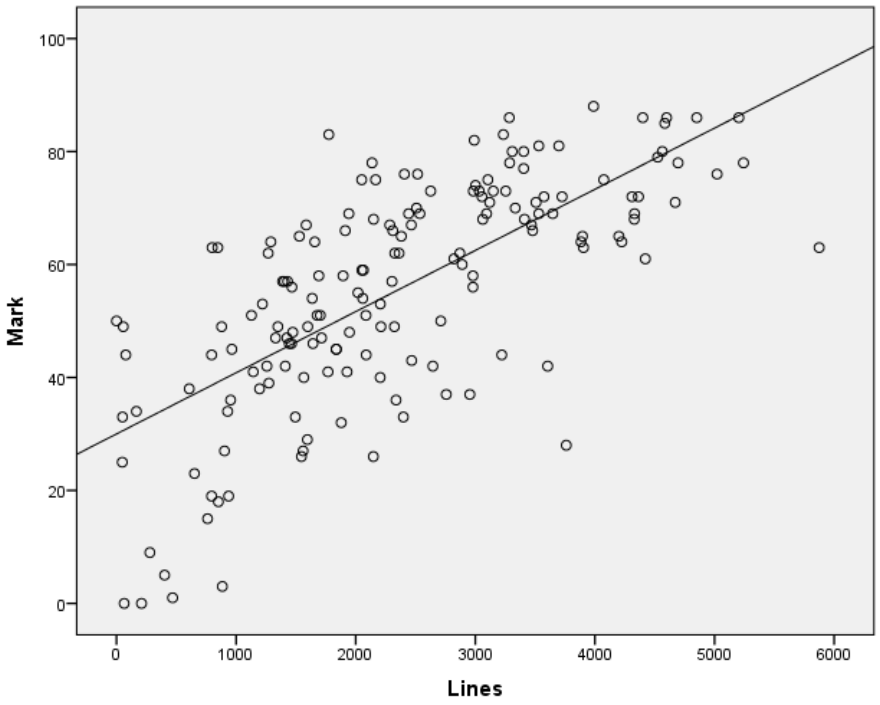


Fig. 2. Lines of Code Changed versus Mark Achieved Without Outliers

It is worth taking some time to consider why these outlier students made so many more changes than their peers. The source control logs revealed very well documented changes, some even using more advanced features of source control such as branching. These students typically worked steadily and consistently for the duration of the module. The code itself was often over-engineered for the scope of the module, indicating that these students were stretching themselves beyond the expectations of their tutors and creating their own challenges.

4 Conclusion and Further Work

Introducing source control to computer science students has clear benefits in terms of their employability as software developers. In education, however, it is proposed that there are additional benefits to using source control. Source control can provide students with a means to low risk experimentation on their code. When needed it can provide them with an easier and more productive route to assistance. It can provide a more efficient and appropriate link between instructors and students for the delivery of code based course material, a natural mechanism for submitting assessment and a means to provide feedback.

Student's use of source control also generates a great deal of data to be mined. The analysis of the data produced is an ongoing process. Future analysis will consider when students made changes both with respect to the working week, and the University trimester, and what changes students made. Although there does not yet seem to be any clear predictors for student performance there may be indicators of students who are failing to engage fully and are at risk of falling behind in their studies. With this goal, data should be extracted from source control for the first few weeks to see if there are any indicators of withdrawal early enough to make a timely intervention.

References

1. Brindescu, C., Codoban, M., Shmarkatiuk, D. D.: How Do Centralized Distributed Version Control Systems Impact Software Changes? ICSE 2014 (2014)
2. Clifton, C., Kaczmarczyk, L., Mrozek, M. Subverting the Fundamentals Sequence: Using Version Control to Enhance Course Management ACM SIGCSE, 39, 1, 86–90 (2007)
3. Assessing Collaborative Undergraduate Student Wikis and SVN with Technology-based Instrumentation: Relating Participation Patterns to Learning American Society of Engineering Education Conference, (2011)
4. Glassy, L.: Using Version Control to Observe Student Software Development Processes. Journal of Computing Sciences in Colleges 21, 3, 99–106 (2006)
5. Gregorio, R., Gonzalez-Barahona, J. Mining Student Repositories to Gain Learning Analytics. An Experience Report Global Engineering Education Conference (EDUCON) (2013)
6. Helmick, M. Integrating Online Courseware for Computer Science Courses ITiCSE'07, ACM Press, 39, 2, 146–150 (2007)
7. Jones, C. Using Subversion as an aid in evaluating individuals working on a group coding project Journal of Computer Sciences in Colleges, 25, 3, 18–23 (2010)
8. Kelleher, J. Employing Git in the Classroom WCCAIS'2014, (2014)
9. Kertész, C. Using GitHub in the Classroom - a Collaborative Learning Experience. SIITME 21, 381–386 (2015)
10. Kim, J., Shaw, E., Xu, H., Adarsh, G. V. Assisting Instructional Assessment of Undergraduate Collaborative Wiki and SVN Activities International Conference on Educational Data Mining (EDM), (2012)
11. Koc, A., Tansel, A.: A Survey of Version Control Systems. ICEME 2011 (2011)
12. Lawrance, J., Jung, S., Wiseman, C. Git on the Cloud in the Classroom SIGCSE'13, ACM Press, 639–644, (2013)

13. Liu, Y., Stroulia, E. Wong, K. Using CVS Historical Information to Understand How Students Develop Software MSR 1, 32–36 (2004)
14. Liu, S., Kim, J., Macskassy, S., Shaw, E. Predicting Group Programming Performance using SVN Activity Traces International Conference on Educations Data Mining, (2013)
15. Ljubovic, L., Nosovic, N. Repository Analysis Tools in Teaching Software Engineering IX International Symposium on Telecommunications (BIHTEL) (2012)
16. Mierle, K., Laven, K., Roweis, S., Wilson, G. Mining Student CVS Repositories for Performance Indicators ACM SIGSOFT Software Engineering Notes, 30, 4, 1–5 (2005)
17. Milentijevic, I., Vlanimir, C., Vojinovic, O. Version Control in Project Based Learning. Computers & Education 50, 1331–1338 (2008)
18. Novák, M. Biňas, M., Michalko, M., Jakab, F. Student's Progress Tracking on Programming Assignments Emerging eLearning Technologies & Applications (ICETA), 279-282 (2012)
19. Reid, K., Wilson, G. Learning by Doing: Introducing Version Control as a Way to Manage Student Assignments SIGCSE'05, ACM Press, 272-276, (2005)
20. Zagalsky, A., Feliciano, J., Storey, M., Zhao, Y., Wang, W. Emergence of GitHub as a collaborative platform for Education CSCW'15, ACM Press, 1906-1917, (2015)

Selecting Execution-Time Server Parameters for Real-Time Stream Processing Systems

HaiTao Mei, Ian Gray and Andy Wellings

University of York, UK
(hm857, Ian.Gray, Andy.Wellings)@york.ac.uk

Abstract. This paper considers the integration of the stream processing programming model into a hard real-time system. It proposes the use of execution-time servers as a mechanism for giving good response times to the stream processing components without compromising the guarantees given to the hard real-time components. Attention is focused on selecting the parameters of the servers so that any spare processor utilisation is dedicated to the stream processing activities at a high priority. A server parameter selection algorithm is presented and evaluated.

Keywords: Real-Time Stream Processing, Server Parameter Selection

1 Introduction

In real-time systems, tasks are often classified as being hard or soft. Hard real-time tasks are those where it is absolutely imperative that responses occur within the specified deadline. Soft real-time tasks are those where response times are important but the system will still function correctly if deadlines are occasionally missed [2].

Due to increased computational demands, modern real-time systems now execute on multiprocessor platforms, and potentially as part of a cluster-based architecture. The stream processing programming model [14] is a programming model that aims to facilitate the construction of concurrent programs to exploit the parallelism available on these architectures. The paradigm is particularly suited to applications that have components that must process large volumes of data; so-called Big Data applications [8]. Although many of these applications have real-time requirements [1] most stream processing architectures are not targeted towards these requirements.

In this paper we address the integration of a stream processing paradigm into a real-time environment that also has hard real-time components. Processing streaming data is often computationally intensive and it is difficult to predict the volume and the cost of processing the data. Hence, we consider the components processing the data to be soft real-time, and thus primarily requiring good response times.

Execution-time servers [2] are a technique that is used in the real-time community to ensure that soft tasks that might demand unbounded CPU time still

exhibit good response times. Servers limit the impact of soft tasks on hard tasks, so that the hard tasks can be guaranteed to meet their deadlines. However, the algorithms needed to determine the number of servers required and their parameters have a time complexity that is exponential [2]. Furthermore they have not previously been used in a stream processing environment. This paper provides three main contributions:

1. A proposal for using execution-time servers for processing real-time streaming data in soft real-time.
2. A fast $O(n^2)$ algorithm for selecting the (sub-optimal) number of servers and their parameters, which maximises the processor utilisation and gives good response times to stream processing activities.
3. Experimental evaluation showing the efficacy of the algorithm.

This paper is structured as follows. Section 2 summarises related work in the area of execution-time servers and the selection of the their parameters. This is followed by an overview of the architecture of the York Stream Processing Framework (SPRY) described in Section 3, which provides the context for this work. Section 4 describes how to generate the servers for handling real-time stream processing tasks, and the proposed heuristic that selects parameters for the generated servers. Section 5 evaluates our heuristic, and finally we give our conclusions. Throughout this paper we use the following notations: C = execution time, D = deadline, T = period, U = utilisation, and P = priority.

2 Related Work of Execution-Time Servers

In real-time systems that consist of both hard and soft components it is essential that the resources required by hard real-time components are reserved so that their deadlines are guaranteed. The term *spare capacity* is used to indicate any resources that are not required by the hard real-time tasks, and so therefore are available for the execution of soft real-time tasks [4]. The problems tackled in [4] are to identify the amount of spare capacity, and to make that capacity available at run-time for the execution of soft tasks. The former is determined using schedulability analysis techniques, and the later is supported by execution-time servers (sometimes called aperiodic or sporadic servers).

Execution-time servers are logical periodic tasks that have a set period, priority and capacity. The capacity is the amount of execution time that the server is allowed to execute during each period. Once the capacity is used, the server must wait for it to be replenished. When a server has capacity it is able to perform computation on behalf of its allocated tasks. The server is logical as it may not exist at run-time but be represented by some underlying Operating System resource reservation protocol.

There are many different types of execution time servers, see [11] for a review. The POSIX standard supports the sporadic server [7, 13]. A sporadic server has a replenishment period, a budget (or capacity), and two priorities: high priority and low priority. When handling aperiodic events, the server executes at the

high priority when it has budget, otherwise it runs at the low priority. When the server runs at the high priority the amount of execution time that has been consumed is subtracted from its budget. The budget consumed is replenished at a time one server period on from its point of consumption.

The deferrable server [7, 13] allows a new logical thread to be introduced at a particular priority level. If the server's period and capacity are appropriately configured, all the periodic tasks in the system remain schedulable even if the server fully consumes its capacity. When registered with a deferrable server, an aperiodic thread executes at the server's priority level until either the capacity is exhausted or it finishes its execution. In the former case, the aperiodic thread is suspended or transferred to a background priority. The capacity of a deferrable server is replenished every period.

One of the main problems with the use of execution-time servers is the calculation of their parameters so that the spare capacity can be effectively used. An investigation into server parameter selection for fixed priority preemptive systems is given by Davis and Burns [3] in the context of hierarchical scheduling. Their work proposed using servers to serve applications that have multiple real-time tasks, with the goal of low overall system utilisation. The paper provided a set of algorithms that determine the optimal value for one server parameter. Davis and Burns illustrate how to use exhaustive search to find optimal servers, and show that the general problem is intractable.

An alternative approach to execution-time servers is to use a slack stealing algorithm [6]. However, this approach has to be used in conjunction with an on-line scheduler and has high overheads.

In this paper, we use deferrable servers because they have a simple run-time representation. However, the type of server is not essential to our approach.

3 The Architecture of the York Stream Processing Framework (SPRY)

A stream processing system consists of a collection of modules that compute in parallel and communicate via channels [14]. Modules can be either *source capturing* (that pass data from a source into the system), *filters* (that perform atomic operations on the data) or *sinks* (that either consume the data or pass it out of the system). Real-time stream processing systems are stream processing systems that have time constraints associated with the processing of data elements as they flow through the system from source to sink. In general, the data sources of stream processing systems can be classified into two types [9]: batched and streaming. A batched data source is where the data is already present in memory, and its content and size will not change during processing. A streaming data source represents data that arrives dynamically, its content and size will change with time.

The York Stream Processing Framework (SPRY) was developed to support a streaming data paradigm for both batched and streaming data sources in real-time, which targets shared memory multiprocessor platforms.

Initially, SPRY integrates Java 8 Streams and the Real-time Specification for Java (RTSJ) to support real-time batched data processing in parallel [9]. The Java 8 Stream API enables pipelined or parallelised processing of data sources with concise code. This work replaces the default Java 8 Streams processing infrastructure with our proposed real-time ForkJoin thread pool, which is illustrated in Figure 1. In a real-time ForkJoin thread pool, one real-time worker thread is created for each processor with a priority, and an optional execution-time server.

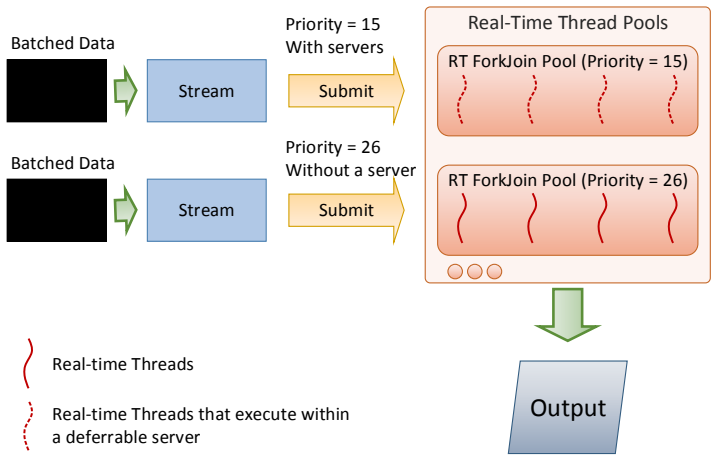


Fig. 1: Overview of the Real-time Stream Framework.

Then, the handling of streaming data in real-time is supported in SPRY by employing our proposed real-time micro batching approach [10]. In a data flow, data items are grouped into micro batches so that each micro batch is treated as a static data source and processed using Java 8 Streams with our real-time ForkJoin thread pool. The real-time streaming facility in SPRY is illustrated by Figure 2.

One goal of SPRY is to process data elements as soon as possible, but also to limit the processing impact on other hard real-time activities in the same

Fig. 2: The overview of real-time processing streaming data.

system by using execution-time servers (e.g. deferrable severs [7]). Currently, SPRY targets fully-partitioned systems, where a task is not allowed to migrate to any other processor once it has been allocated. SPRY pins each worker thread in the real-time ForkJoin thread pool to different processors to support this scheduling scheme.

We assume that real-time activity has already been partitioned via some task allocation approach that takes into account locality of streaming data and its interaction with hard real-time activities. On each processor, the response time of processing each partition of a batched data source can be analysed using current analysis techniques for deferrable servers [3].

4 Selecting Server Parameters for Real-Time Stream Processing in SPRY

In this section, the server parameter selection problem for the SPRY framework is considered. The overall problem is how to find one or multiple deferrable servers against a task set, so that the system utilisation is maximised, and all the hard real-time tasks in the task set remain schedulable. System utilisation is given by the following equation:

$$U_{System} = \sum_{\forall \tau_i \in System} \frac{C_i}{T_i}$$

Typically, deferrable servers require three parameters: priority, period, and capacity. However, in order to make the problem tractable, we fix one of the parameters. Server priority is fixed to the highest priority in the system. The reason for this is that soft real-time tasks must provide a fast response time without being unduly interfered with by hard real-time activities. Selecting a high priority but then limiting the amount of allowed execution time makes the latency of each item in the stream as small as possible whilst maintaining hard real-time guarantees.

Arguably, very long-running batched data processing tasks could be executed at background priority to simply use all available idle time. However, running at high priority introduces benefits when the processing time is relatively short, or partial results are time sensitive.

4.1 Execution-Time Servers Generation

Given a hard real-time periodic task set where each task has a unique priority (e.g., assigned by the deadline monotonic priority assignment algorithm); the proposed approach is described as follows:

1. Put all tasks in the task set into a queue.
2. Get the task from the queue with highest priority as τ_p, the priority of which is p. Remove task τ_p from the queue.

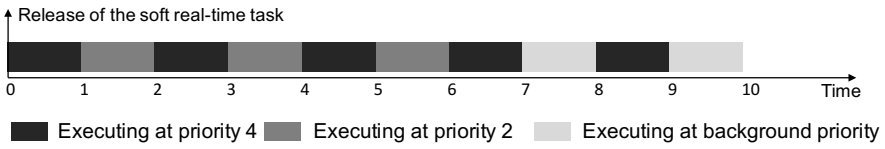


Fig. 3: Execution of the soft real-time task.

3. Create a deferrable server at priority $S = p + 1$.
4. Find a combination of its replenishment period T_S and capacity C_S, so that $\frac{T_S}{C_S}$ is as high as possible, and all the lower priority (i.e., less than or equal to p) tasks in the tasks remain schedulable. This can be done using the proposed heuristic that will be proposed in Section 4.2.
5. Go to step 2 when the queue is not empty, otherwise exit.

For example, consider a task set which consists of two tasks: τ_5 ($C_5 = 3$, $D_5 = 5$, $T_5 = 10$), and τ_3 ($C_3 = 2$, $D_3 = 10$, $T_3 = 10$). The priority 5 is the highest priority. Using the proposed algorithm, two deferrable servers can be found. They are S_1 (with priority = 6, $C_{S_1} = 2$, $T_{S_1} = 10$), and S_2 (with priority = 4, $C_{S_2} = 3$, $T_{S_2} = 10$). After adding these two servers to the system, τ_5 and τ_3 are still schedulable, with R_5=5 and $R_3 = 10$ which can be calculated using the response time analysis (RTA) equations [2].

A set of deferrable servers at different priority levels have been found using the proposed algorithm. At runtime, if the set is not empty, the soft real-time task will be executed by the highest priority deferrable server which has capacity left. When the current capacity is exhausted, the soft real-time task will be transferred to another deferrable server which has the highest priority amongst all servers that have capacity left (if any). Otherwise, the soft real-time task has to be suspended or, more typically, executed at the background priority. The soft real-time task will be transferred to a higher priority server (compared to its current executing priority), when the server's capacity is replenished.

For example, we have two deferrable servers: S_1 (priority = 4, $C_{S_1} = 1$, $T_{S_1} = 2$), S_2 (priority = 2, $C_{S_2} = 3$, $T_{S_2} = 10$), and one soft real-time task which requests 10 time units of execution. They are all released at time 0 and the execution is illustrated by Figure 3.

4.2 Generating Parameters for each Server

The priority of a server is determined using the algorithm proposed in Section 4.1. The period and capacity are yet to be calculated. However, once the period and the priority of a server can be determined, we can employ the optimal server capacity allocation algorithm that was proposed in [3] to calculate the capacity. The proposed capacity searching algorithm uses a binary search to search the capacity between 0 and the period of the server for the maximum capacity C for which all the tasks in the system remain schedulable.

Therefore, the problem then becomes how to choose the period for the server at each priority level.

Difficulties in Server Parameter Selection

Searching for optimal periods for servers at different priority levels has an exponential time complexity. For a task set of n tasks with periods between 0 and p, the complexity of an exhaustive search is n^p. For example, if we try to find servers against a task set that contains 100 tasks where a potential period of each server is from 0 to 10, it will require 10^{100} runs of the binary capacity search. If each run takes 1 nanosecond, the whole search will take about 3×10^{83} years.

It is therefore necessary to employ a discontinuous landscape search algorithm to search towards the optimal server parameters.

Heuristic for Server Parameter Selection

In order to mitigate the time complexity of the server parameter selection problem, we propose a fast and simple heuristic for determining the potential period and capacity for the servers. We use a discontinuous greedy search approach. In the algorithm below, *task* refers to the hard real-time tasks of the system, and *server* to the execution-time servers used to execute the streaming jobs. The algorithm is described as follows:

```
1 Sort servers as highest priority first;
2 Calculate the exact divisors of the deadline of each hard task in the
     task set, and add them into the list: potentialPeriods. The list is
     ordered by longest period first, and contains no duplicate periods;
3 foreach(Server s in servers){
4   ServerUtilisation = 0;
5   foreach(Period t in potentialPeriods){
6      Binary search capacities between 0 and the period of s for the
           maximum capacity C, such that all the hard tasks in the system
           remain schedulable;
7      if(a schedulable capacity found){
8         if(C/t > ServerUtilisation){
9            assign the C as the capacity of server: s;
10           assign the t as the period of server: s;
11           ServerUtilisation = C/t;
12        }
13     }
14  }
15 }
```

Using this algorithm, a proposed set of deferrable servers that are described in Section 4.1 can be found, which is sub-optimal. Given a task set that consists of n hard real-time tasks, the time complexity of this heuristic search approach is $O(n^2)$. This is because the RTA [2] for the schedulability test that is invoked in the `foreach` loop within the heuristic has a $O(n)$ complexity. Even still, the

algorithm is very fast. For example, on a personal laptop computer, it only takes 25 seconds to find servers for a task set of 100 tasks with maximum period 100. The performance of this algorithm is evaluated in Section 5.

5 Evaluation of Server Parameter Selection Algorithm

This section presents the results of empirical investigations into our heuristic for server parameter selection.

The experiment evaluates the performance of our heuristic against schedulable task sets with total utilisations of 30%, 40%, 60%, and 80%. At each utilisation level the task set size varies from 1 to 100 and the total utilisation is normally distributed into each task using the algorithm proposed in [5].

Synthetic task sets are generated for the experiments. For each task within a task set, the period is a unique randomly generated number between 10 and 1000 and the execution-time is calculated using $C_i = U_i \times T_i$, where U_i and T_i represent the utilisation and the period of the task. The schedulability of each generated task set is validated using response time analysis proposed in [2]. Within each task set, the deadline of each task is equal to its period. Experiments are performed 100 times and the results are shown in Figure 4.

The results are presented in box plot graphs of the system utilisation achieved after having applied the determined server parameters. This is given by:

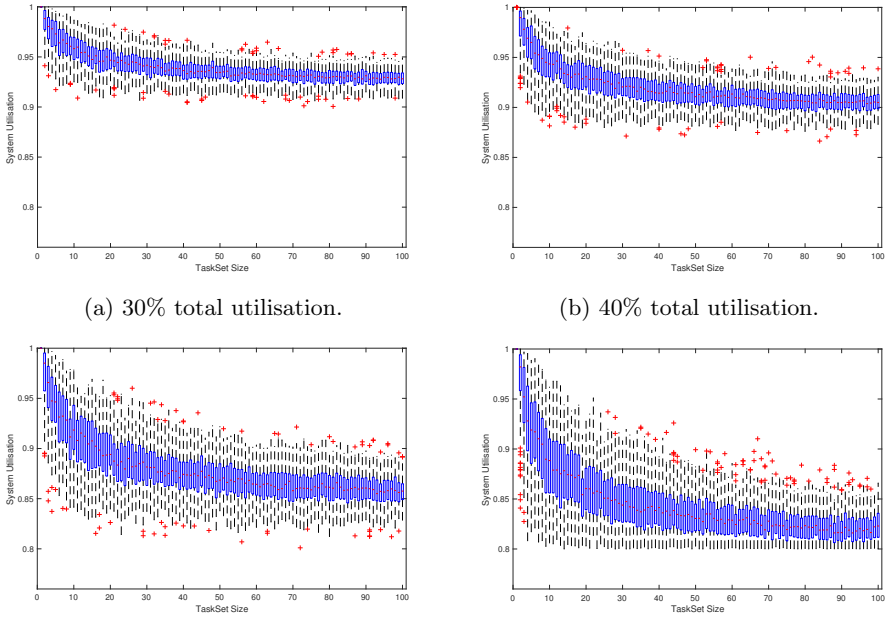
$$\sum_{\forall S \in Servers} \frac{C_S}{T_S} + \sum_{\forall \tau_i \in Taskset} \frac{C_i}{T_i}$$

System utilisation is shown on the y-axis, whilst the size of the task set is on the x-axis.

Overall, our heuristic improves the system utilisation significantly. When the utilisation of the task set is 30%, the system utilisation is improved to above 94% on average for arbitrary task set size. The system utilisation achieves 100% when the task set has only one task, this is because the algorithm simply creates a server with the same period as the task and using all available slack (i.e., $D_i - C_i$, $D_i = T_i$) of the task as its capacity.

The mean system utilisation decreases slightly when the task set utilisation increases, this is because the remaining space for the server is decreased. For example, given a task set with $\tau_1(D_1 = T_1 = 30, C_1 = 5)$, and $\tau_2(D_2 = T_2 = 20, C_2 = 5)$, we can have a server $(T_S = 4, C_S = 1.857)$ resulting in system utilisation of 88.1%. If C_1 and C_2 are set to 10 then no server can be found and system utilisation remains 83.3%.

In addition, the mean system utilisation also decreases when the size of the task set increases. The reason is that the more tasks, the lower the chance that they display harmonic periods. A task set with fully harmonic periods enables 100% utilisation [12]. For example, if we have two tasks with harmonic periods: $\tau_1(D_1 = T_1 = 40)$, $\tau_2(D_2 = T_2 = 20)$, and the utilisation of each is 25%, we can find a server $(T_S = 8, C_S = 4)$ which makes system utilisation 100%.



(a) 30% total utilisation.

(b) 40% total utilisation.

(c) 60% total utilisation.

(d) 80% total utilisation.

Fig. 4: System utilisation after using servers for task sets with deadline equal to period.

However, if we reconfigure T_1 to 30, maximum system utilisation can be shown through exhaustive searching to be only 89.3%. As discussed previously, exhaustive searching is only feasible for the very smallest task sets.

It can be observed that as the utilisation of the hard real-time tasks increases, the total utilisation achievable by this approach varies increasingly. This is because our approach only uses a single execution server for the task set with the deadline equals to the period, which must be placed at a single priority level. As the utilisation increases, it is harder to find a space in the schedule to efficiently use the available CPU time.

6 Conclusions

In this paper, we have investigated how to use execution-time servers for real-time stream processing in fixed priority-based pre-emptive multiprocessor systems. The proposed approach is to use a set of deferrable servers to execute the soft real-time stream processing tasks at as high a priority as possible, whilst still guaranteeing the hard real-time tasks in the system remain schedulable. A heuristic has been proposed for server parameter selection that works for tasks with their deadline less than or equals to their periods. The experiments show

that significant utilisation can be made available for handling stream processing tasks when using our heuristic.

References

1. N. C. Audsley, Y. Chan, I. Gray, and A. J. Wellings. Real-Time Big Data: the JUNIPER Approach. 2014.
2. A. Burns and A. Wellings. *Real-time systems and programming languages: Ada 95, real-time Java, and real-time POSIX*. Pearson Education, 2001.
3. R. Davis and A. Burns. An investigation into server parameter selection for hierarchical fixed priority pre-emptive systems. In *16th International Conference on Real-Time and Network Systems (RTNS 2008)*, 2008.
4. R. I. Davis. *On exploiting spare capacity in hard real-time systems.* PhD thesis, University of York, 1995.
5. R. I. Davis and A. Burns. Improved priority assignment for global fixed priority pre-emptive scheduling in multiprocessor real-time systems. *Real-Time Systems*, 47(1):1–40, 2011.
6. R. I. Davis, K. W. Tindell, and A. Burns. Scheduling slack time in fixed priority pre-emptive systems. In *Proceedings of the Real-Time Systems Symposium, 1993.*, pages 222–231. IEEE, 1993.
7. J. P. Lehoczky, L. Sha, and J. K. Strosnider. Enhanced Aperiodic Responsiveness in a Hard Real-Time Environment. In *Proceedings of the IEEE Real-Time Systems Symposium*, pages 261–270, 1987.
8. J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers. Big data: The next frontier for innovation, competition, and productivity. 2011.
9. H. Mei, I. Gray, and A. Wellings. Integrating Java 8 Streams with The Real-Time Specification for Java. In *Proceedings of the 13th International Workshop on Java Technologies for Real-time and Embedded Systems*, page 10. ACM, 2015.
10. H. Mei, I. Gray, and A. Wellings. Real-Time Stream Processing in Java. In *Ada-Europe International Conference on Reliable Software Technologies*, pages 44–57. Springer, 2016.
11. L. Sha and J. B. Goodenough. Real-time scheduling theory and Ada. Technical report, DTIC Document, 1989.
12. L. Sha, M. H. Klein, and J. B. Goodenough. Rate monotonic analysis for real-time systems. In *Foundations of Real-Time Computing: Scheduling and Resource Management*, pages 129–155. Springer, 1991.
13. B. Sprunt, L. Sha, and J. P. Lehoczky. Aperiodic Task Scheduling for Hard Real-Time Systems. *Real-Time Systems*, 1:27–69, 1989.
14. R. Stephens. A Survey of Stream Processing. *Acta Informatica*, 34:491–541, 1997.

Analysing Fish Behaviours Using Three-Dimensional Qualitative Trajectory Calculus

Alaa AlZoubi[1], Patrick Dickinson[1], Thomas W. Pike[2], and Bashir Al-Diri[1]

[1] School of Computer Science, University of Lincoln, Lincoln, UK
[2] School of Life Sciences, University of Lincoln, Lincoln, UK
{aalzoubi,pdickinson,tpike,baldiri}@lincoln.ac.uk

Abstract. Fish swim freely in water and their 3D spatial interactions may carry important information for biological study. The two-dimensional Qualitative Trajectory Calculus (QTC_{2D}), a spatiotemporal calculus, is a method for representing and reasoning about movements of objects in a qualitative framework. This emerging technique encodes the spatial geometric relationships between distinct objects, such as machines, humans or animals. This paper presents a method for generalising QTC_{2D} into 3D space, called $3DQTC$, as a work-in-progress. The $3DQTC$ method is based on a geometrical analysis method of estimating the 3D orientation of fish. Initial results indicate the potential of representing and analysing the fish behaviours in 3D space. $3DQTC$ is also demonstrated as a means of extracting useful information (e.g. *follow* and *non-follow* behaviours) which cannot be achieved by QTC_{2D}. We conclude by discussing further work and development.

1 Introduction

The development of computer vision as a method for automatically analysing human activities is a well-established research area. However, applications to the analysis of animal behaviours are relatively rare. The three-spined stickleback fish (*Gasterosteus aculeatus*) is a model species, used for behavioural studies in laboratories around the world [1]. These species are widely used for studies in behavioural ecology [1] as they are relatively easy to collect, to house and manipulate in the laboratories. Stickleback fish have similar shape, colour and size which makes the tracking and identification process a challenge. Current methods of studying their social behaviours rely on manual observation and recording, which is very time-consuming, error prone, limiting on the amount of data which can be collected, and applicable only to small number of fish.

Qualitative Trajectory Calculus (QTC) is "a calculus for representing and reasoning about movements of objects in a qualitative framework" [2]. The QTC was developed to represent the relative movements between two disjoint objects in 1D and 2D space. In our current work, QTC_{2D} provides a basis of representing and analysing fish movements and interactions.

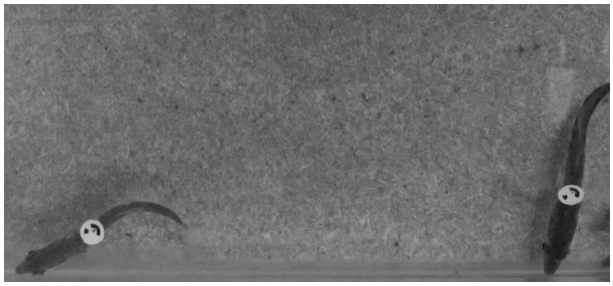


Fig. 1. Pair of three-spined sticklebacks with a unique circular tags.

Existing methods apply the QTC to represent the spatial interactions between moving objects in 2D space. In this paper, we present our new method for generalising Qualitative Trajectory Calculus into 3D Space (called $3DQTC$) based on the geometric method of estimating the 3D orientation, as presented in [3]. We combine the geometrical analysis method [4] and our Reprojection method [3] to define the three-dimensional interactions between a pair of fish. Then, a new variant of QTC ($3DQTC$) was constructed based on the 3D information. The aim of this work is to extract useful information which cannot be achieved by QTC_{2D} such as: pairs of fish swim on same or different water surface. Our system, and for the first time, represent and capture the fish relative movements in 3D space and their behaviours using qualitative representations. We also present a novel fish identification, tracking and analysis method in 3D space which automatically provides accurate measurements for biologists studying behaviour of these fish, providing much larger and more robust data sets than can be gathered manually. We use a circular marker system (tagging system) presented in [1, 3] to identify individuals; these tags are attached to the dorsal spine, and our system estimates their position and orientation over a period of time. Each circular marker (tag) has a unique pattern. Fig. 1 shows a sample image of three-spined sticklebacks.

2 Related Work

2.1 Fish Monitoring

There has been some previous work describing video-based systems for automated (or semi-automated) processing of fish behaviour. Zhu [5] studied a stereo-vision based real-time tracking method to monitor the 3D behaviour of aquatic animals. However, this method is used over relatively short time periods only. A model-based approach, which estimates the dimensions of free-swimming fish, was developed in [6], but requires manual operator inputs. A machine vision system that automatically analyses underwater videos for counting fish was presented in [7], while Kane et al. [8] developed a movement analysis system

to measure variables such as velocity, distance and space utilisation of fish in tanks. An online monitoring system for fish behaviour was presented in [9]; the system detected abnormal behaviour. However, the methods in [8, 9] are either limited to the tracking a single fish, or are only applicable to relatively short time periods ($<$ 15 minutes). Recently, the method in [10] investigated individual level interactions between shoaling sticklebacks. The method manually estimates the 2D positions of sticklebacks to analysis their interactions. This method is time-consuming, error prone, and the amount of data which can be collected is limited.

2.2 Qualitative Spatial and Temporal Reasoning

Qualitative spatial-temporal reasoning is an approach for dealing with knowledge on which human perception of relative interactions is based without using numerical computation [2]. There has been previous work introduced in qualitative spatial and temporal calculi such as Qualitative Trajectory Calculus (QTC) [2]. The QTC describes and encodes the interactions between Moving Point Objects (MPOs) in a qualitative way. It reduces the continuum to three qualitative values (or symbols) -, 0 and +. In the case where the changing in the distance between two MPOs is considered; the symbol - means a decrease in distance between both objects, + an increase in distance and 0 if the distance remains the same.

Different types of QTC have been proposed depending on the level of details and the number of spatial dimensions: QTC Basic (QTC_B) considers only the changing distance between two MPOs and QTC Double Cross (QTC_C) consider the direction in which an object is moving with respect to the segment line connecting between the two objects. Given the positions of two moving objects (called Obj_1 and Obj_2); the QTC represents the relative motion between the two objects at instant time as follow:

1. $Code_1$: movement of Obj_1 with respect to Obj_2
   - : Obj_1 approaching Obj_2
   + : Obj_1 moving further away from Obj_2
   0 : Obj_1 is stable with respect to Obj_2
2. $Code_2$: movement of Obj_2 with respect to Obj_1
   similar to $Code_1$ but with the Obj_1 and Obj_2 swapped.
3. $Code_3$: Relative speed of Obj_1 with respect to Obj_2 (which dually represents the relative speed of Obj_2 with respect to Obj_1):
   - : Obj_1 slower than Obj_2
   + : Obj_1 faster than Obj_2
   0 : Obj_1 and Obj_2 move with the same speed
4. $Code_4$: Movement of Obj_1 with respect to the reference line L
   - : Obj_1 moves to the left of L
   + : Obj_1 moves to the right of L
   0 : Obj_1 move a long the line L
5. $Code_5$: Movement of Obj_2 with respect to the reference line L
   similar to $Code_4$ but with the Obj_1 and Obj_2 swapped.

6. $Code_6$: θ_1 the minimal angle between the velocity vector of Obj_1 and vector L, and θ_2 the equivalent for Obj_2
   - : $\theta_1 < \theta_2$
   + : $\theta_1 > \theta_2$
   0 : $\theta_1 = \theta_2$

where $Code_i$ represents the qualitative relations in QTC and L is the line connecting between the two objects at time t_1. Fig. 2(a) illustrates the concept of qualitative relations for two disjoint objects. The two types, QTC_B and QTC_C, have been defined into different subsets as follow: QTC_{B21} contains ($Code_1$ and $Code_2$) and QTC_{B22} contains ($Code_1$, $Code_2$ and $Code_3$). QTC_{C21} contains ($Code_1$, $Code_2$, $Code_4$ and $Code_5$) and QTC_{C22} contains $Code_1$ through $Code_6$. The combinations of the six codes results in $3^6 = 729$ QTC states, whereas QTC_{C22} has only 305 possible states [2]. There has been some applications for QTC such as: moving vehicles [11] and human-robot interaction [12]. A detailed description for QTC relations in the domain of fish interactions is presented in Section 3.

Most of the existing methods and applications of QTC are in 2D space. However, one existing work has been presented for QTC in 3D space by Mavridis et al. [13]. The method developed in [13] generalises QTC into 3D space based on transformations of the Frenet-Serret frames. Two Frenet-Serret frames (t_1, t_2) are used to represent the two moving points. Each frame consists of *tangent (t), normal (n)*, and *binormal (b)* vectors. The method calculate the Euler angles $A_{ng} \in (yaw, pitch,$ and $roll)$. Then, the three Euler angles are mapped into qualitative symbols $\{-, 0, +\}$. In [13], this process is applied for modeling bird flight in 3D space. However, this method has a limitation in real applications and from mathematical point of view: the Fernet frame has an inherent drawback in that it is undefined at points where the curvature is zero [14]. Therefore, when the object moves in a straight-line (collinear curve) or remains stationary, the method will fail in representing the relative movement. Moving in a straight-line or staying stationary is a common state in object behaviours such as fish. For example, the "Approach" fish behaviour, where one fish moves toward another fish while the other fish standing still.

3 Representing Fish Behaviour Using QTC_{2D}

The use of qualitative spatial representations is an adequate and powerful method to abstract a large number of possible objects interactions (e.g. fish) or scenarios such as "one fish follow another fish". In this section we introduce QTC_{2D} as a method for representing the spatial interactions between a pair of fish. We have developed an automated visual tracking method which estimates the pose of the tags (Fig. 1), fish position and orientation, from monocular video. Our method is fully described in [3], and comprises the following components:

Camera Calibration: The camera parameters (focal length f, principle point of the image plane and lens distortion factors) are estimated using [15].

Image Enhancement: An adaptive background mixture model with shadow

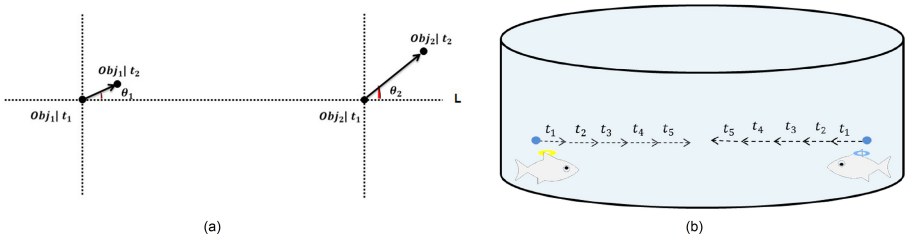


Fig. 2. (a) Example of QTC relations between Obj_1 and Obj_2. (b) Example of spatial interactions between two fish (converge behaviour) during the time interval t_1 to t_5.

detection [16] is used to locate moving objects and eliminate the image noise.
Circular Feature Edge Detection: The Canny edge detector method [17] is used to detect elliptical shapes of the circular tag as a set of pixel edge points.
Lens Distortion Compensation: Radial lens distortion factors obtained in camera calibration process [15] are used to find accurate positions of edge points.
2D Position Estimation: The direct least square ellipse fitting method [18] is applied to estimate the basic parameters of the elliptical projection of the tag. The center of the tag represents 2D fish position.
3D Orientation Estimation: Our Reprojection method [3] is applied to estimate the 3D orientation of the circular tag. Fig. 3, shows the schematic representation of 3D orientation estimation of fish swimming.
Tag Identification: The method in [1] is used to read the tag pattern (fish ID).

We adopt QTC_{2D} to represent and reason about fish movements in a free Euclidean space. The spatial position of the fish movement is abstracted to a single point, which represents the tag centroid. As an example, consider the interaction between two fish during a given time interval as shown in Fig. 2(b), which represents a section of real data from our dataset. The interaction is captured using QTC representations: both fish are converging during the time interval $[t_1, t_5]$. This interaction is described as follow using QTC state sequences. Using QTC_{B21}: $(-\ -)_{t1} \rightsquigarrow (-\ -)_{t2} \rightsquigarrow (-\ -)_{t3} \rightsquigarrow (-\ -)_{t4} \rightsquigarrow (-\ -)_{t5}$. The QTC_{C21}: $(-\ -\ 0\ 0)_{t1} \rightsquigarrow (-\ -\ 0\ 0)_{t2} \rightsquigarrow (-\ -\ 0\ 0)_{t3} \rightsquigarrow (-\ -\ 0\ 0)_{t4} \rightsquigarrow (-\ -\ 0\ 0)_{t5}$. The interaction may also be described with more details using QTC_{C22} by including the speed and angle features, but the expansion is omitted.

4 Three-Dimensional QTC

This section presents our novel method (called $3DQTC$) of generalising QTC into 3D space. In Section 2.2, we defined four QTC_{2D} features which map onto the six codes: distance:$\{Code_1, Code_2\}$; side:$\{Code_4, Code_5\}$; speed:$\{Code_3\}$; and angle:$\{Code_6\}$. The distance, speed and angle features have equivalents in 3D space, and can be easily generalised by defining these features in 3D space. However, there is no equivalent side feature ($Code_4, Code_5$). The main challenge of generalising 2D QTC into 3D QTC is that in 2D space a unique line connecting

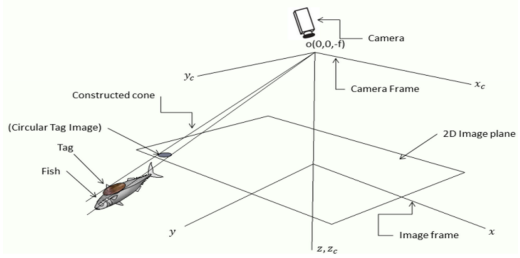


Fig. 3. Schematic representation of 3D orientation estimation of fish swimming.

the two moving points can be drawn, and a plane which define the left/right regions (Fig. 2(a)). However, such a unique plane cannot be defined between two points in 3D space. We propose to use the geometrical analysis method [4] and our Reprojection method [3] to define the direction movement between a pair of fish in 3D space. We use our Reprojection method [3] to estimate the 3D orientation (α_j, β_j, and γ_j) of each fish. This orientation values are then used to represent the analogous *side* feature (called *orientation feature*) in 3D space. The *orientation feature* will encode the relative change in the three angles α_j, β_j, and γ_j of a pair of fish.

Definition: Given the 3D orientations (V_1, V_2) and positions (P_1, P_2) of two moving objects in 3D space (called Obj_1 and Obj_2), where $V_1 = [\alpha_1, \beta_1, \gamma_1]^T$ and $V_2 = [\alpha_2, \beta_2, \gamma_2]^T$ are the 3D orientations of Obj_1 and Obj_2, respectively. The $3DQTC$ represents the relative motion between Obj_1 and Obj_2 at time instant t as follows:
First, we define the distance:$\{Code_1, Code_2\}$; speed:$\{Code_3\}$; and angle:$\{Code_6\}$ features similarly to QTC_{2D} generalised into 3D space. We calculate the symbols $\{-, 0, +\}$ for all three features. Secondly, we define a new feature called "*orientation feature*" as follows:
orientation feature: The relative change of the orientation between Obj_1 and Obj_2 can be defined as the differences $d_\alpha = (\alpha_1 - \alpha_2); d_\beta = (\beta_1 - \beta_2)$; and $d_\gamma = (\gamma_1 - \gamma_2)$ between V_1 and V_2 components. This can be described using new three codes ($Code_7, Code_8, Code_9$) as follows:

7. $Code_7$:	8. $Code_8$:	9. $Code_9$:
$-: d_\alpha < -\kappa$	$-: d_\beta < -\kappa$	$-: d_\gamma < -\kappa$
$+: d_\alpha > \kappa$	$+: d_\beta > \kappa$	$+: d_\gamma > \kappa$
$0: -\kappa \leq d_\alpha \leq \kappa$	$0: -\kappa \leq d_\beta \leq \kappa$	$0: -\kappa \leq d_\gamma \leq \kappa$

where $Code_i$ represents the qualitative relations in $3DQTC$, and κ is a threshold. In the new variant of QTC ($3DQTC$) there exist 7 codes ($Code_1, Code_2, Code_3$, $Code_6, Code_7, Code_8, Code_9$) represented in $3^7 = 2187$ combinations of symbols, and $3^3 = 27$ possible combinations of symbols for the *orientation feature*.

Note that the codes: $Code_1$, $Code_2$, $Code_3$, $Code_6$ are analogous to the ones in QTC_{2D}. On the other hand, the three codes: $Code_7$, $Code_8$, $Code_9$ are the ones differentiate between $3DQTC$ from QTC_{2D}. Therefore, in our fish experiment, the focus will be on using these triple features to capture fish interactions which can not be captured using QTC_{2D}.

Table 1. QTC_{B21} relations occurrences and duration for two fish in the video.

QTC Relation	Occurrences	Duration(s)
(- +)	12170	811.4
(+ -)	4604	307
(0 -)	4943	198.3
(- -)	3564	237.7
(- 0)	4943	329.6
(0 +)	2742	182.9
(+ +)	3200	213.4
(+ 0)	3695	246.4
(0 0)	16109	1074

Table 2. Fish behaviours classifications based on QTC_{B21} relations.

group #	QTC Relation	Behaviour
group 1	(- +), (+ -)	Follow
group 2	(0 +), (+ +), (+ 0)	Diverge
group 3	(0 -), (- -), (- 0)	Converge
group 4	(0 0)	Stationarity

5 Experiments

We evaluate the effectiveness of our proposed system and our novel $3DQTC$ in this section. The experiment setup utilised the following hardware imaging components: A canon camera (PowerShot SX200) with resolution 1280×720 pixels and focal length: 5-60mm f/3.4-5.3. A $30cm \times 20cm$ calibration board containing 6×6 equal squares used for camera calibration process. A black circle tank ($30cm$ diameter and $15cm$ water depth) is used to house the fish. An Intel Core i5-2450M laptop, CPU@2.50GHz was used to run the experiments.

5.1 Experiment I: Relative Fish Movements

Our dataset includes one hour of video for two fish captured (with rate 30 frames per second) using the hardware imaging components described in section 5. This dataset was used as an input for our system to analyse fish spatial behaviours. In this experiment, we focus on the QTC_{B21} relations between two fish. The computational procedures of our framework in Section 3 were applied to estimate the position and orientation for each circular tag (fish position and orientation). Then, the QTC_{B21} codes were extracted.
Table 1 displays results pertaining to the relations and interactions between the two fish in the whole dataset. It summarises the total number of occurrences and the total duration for each relation for all fish interactions. It also shows that all nine QTC_{B21} relations do have at least one occurrence, and the occurrences

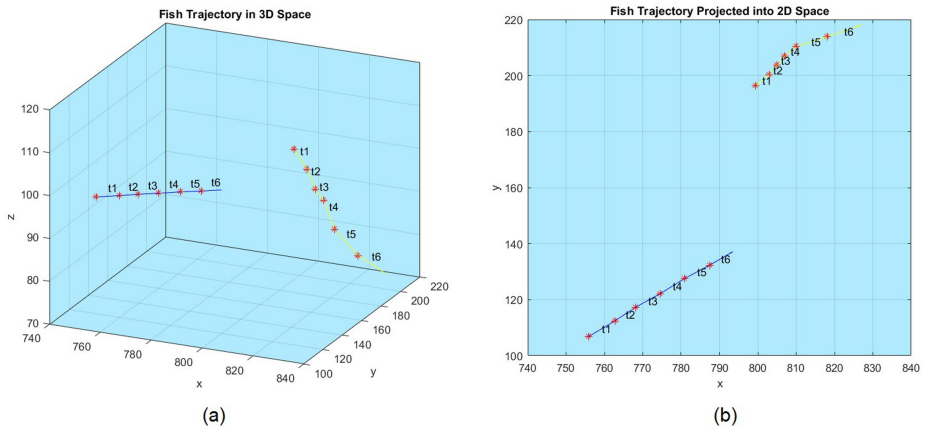


Fig. 4. (a) Representation of a pair of fish trajectory in 3D space. (b) Same trajectory in (a) projected into 2D space.

of these relations in the dataset are not equally distributed. Based on the biologist classifications for these nine fish interactions, four groups of behaviours (Table 2), were extracted as follows: The first group represent the $(-\ +)_{B21}$ and $(+\ -)_{B21}$ relations. It is the most common relations occur between the two fish in the dataset, lasting for around 0.33 of the cumulative time. This indicates that the two fish perform a *follow* behaviours; where the $(-\ +)_{B21}$ relation occurs at 0.24; Obj_1 moves toward Obj_2 and Obj_2 moves away. On the other hand, the relation $(+\ -)_{B21}$ represents the case where Obj_2 moves toward Obj_1 and Obj_1 moves away, occurs at just 0.09. The second group represent the $(0\ +)_{B21}$, $(+\ +)_{B21}$ and $(+\ 0)_{B21}$ relations. These relations occurs 0.18 of the time and represent the *diverge* behaviours. The relations $(0\ -)_{B21}$, $(-\ -)_{B21}$ and $(-\ 0)_{B21}$ represent the third group, where the two fish perform a *converge* behaviours; both fish are approaching each other or one of them approaching and the other is stationarity. Finally, the fourth group is represented by the symmetric $(0\ 0)_{B1}$ relation; which imply that the two fish are *stationarity*. Table 2 shows the biological categorisations of the QTC_{B21} relations.

5.2 Experiment II: Fish "Follow" Behaviour in 3D Space

To test the feasibility of our $3DQTC$, the *follow* behaviour have been chosen from our dataset described in Section 5.1. We selected a subset of our data where Obj_1 *follow* Obj_2 (group 1: $(-\ +)_{B21}$). We estimated the 3D orientation of each individual in this subset. The fish orientation represents the rotation angles that the surface normal vector of the circular tag makes with x, y and z axes (camera frame), respectively. Note that the z axes of the camera frame is perpendicular to the water surface. Then, we used the estimated $\alpha_j, \beta_j, \gamma_j$ pairs for each fish as input and a threshold $\kappa = 5°$, and constructed corresponding $3DQTC$ relations (*orientation feature*: $\{Code_7, Code_8, Code_9\}$) using our method described in

Section 4. The threshold κ has been defined experimentally, and we have found $\kappa = 5°$ is adequate. We used the combinations of three codes ($3^3 = 27$ relation) as a feature to derive a meaningful qualitative representation for fish interaction. Finally, we extracted the total number of occurrences and the total duration for each *orientation feature* relations. This results in splitting group 1 (*follow* behaviour) into two groups, the first group 1.1 represents the (0 0 0) relation and shapes 91% of the dataset. This represents the cases where the two fish swim on same or parallel water surface (insignificant variations in α, β, γ angles). While the second group 1.2 (shapes 9% of the dataset) contains all other cases where the two fish swim in different water surface.

An example of two sequences of $3DQTC$ relations (*orientation feature*) in 6 consecutive observations are as follow: $S_1\{0\,0\,0\,, 0\,0\,0\,, 0\,0\,0\,, 0\,0\,0\,, 0\,0\,0\,, 0\,0\,0\}$; $S_2\{-\,+\,-\,, -\,+\,-\,, -\,+\,-\,, -\,+\,0\,, -\,+\,-\,, -\,+\,-\}$. Note that in S_1 both fish swim on same or parallel surface and according to biological experts, the two fish perform *follow* behaviour. While the fish swim in different water surface in S_2, which is *non-follow* behaviour. Fig. 4 shows that the two fish swim on different water surface and our $3DQTC$ capture this information. However, QTC_{2D} in Experiment I classify these relations as *follow* behaviour. This information is very important for biological study which we could not capture using QTC_{2D}.

6 Conclusion and Discussion

A new method for generalising 2D QTC into 3D space called $3DQTC$ was presented. Our method uses the geometrical analysis, the Reprojection methods and our fish recognition and tracking system to define the three-dimensional interactions between pairs of fish. We applied our method on a real-word fish dataset (*follow* behaviour). We show that our $3DQTC$ coding scheme can provide rich representations of spatial interactions between pairs of fish in 3D space, and capture information of high value for biological study which can not be represented using QTC_{2D}. Our method solves the cases where the objects are stationary or moving in straight-line. A QTC calculi for representing and analysing the spatial behaviours of fish in 2D space has been also presented. QTC_{2D} gave us the benefit of qualitative abstraction defined the fish states.

Encouraged by our initial results, we plan to extend our work in a number of ways. We intend to further to implement the full $3DQTC$ (seven codes: $Code_1$, $Code_2$, $Code_3$, $Code_6$, $Code_7$, $Code_8$, $Code_9$). This results in $3^7 = 2187$ states, and require to define the possible states in real-life. We plan to encode the fish interactions as a single trajectory of $3DQTC$ states. Then, apply standard time-series analysis methods to group the fish behaviours. We also intend to develop a method to analyse differences in behaviour between pairs of fish in 3D space, over long-run datasets.

References

1. T. Kleinhappel, A. Al-Zoubi, B. Al-Diri, O. Burman, P. Dickinson, L. John, A. Wilkinson, and T. Pike, “A method for the automated long-term monitoring

of three-spined stickleback gasterosteus aculeatus shoal dynamics," *Journal of fish biology*, vol. 84, no. 4, pp. 1228–1233, 2014.
2. N. Van de Weghe, "Representing and reasoning about moving objects: A qualitative approach," Ph.D. dissertation, Ghent University, 2004.
3. A. AlZoubi, T. K. Kleinhappel, T. W. Pike, B. Al-Diri, and P. Dickinson, "Solving orientation duality for 3d circular features using monocular vision," in *Proceedings of the 10th International Conference on Computer Vision Theory and Applications*, 2015, pp. 213–219.
4. R. Safaee-Rad, I. Tchoukanov, K. C. Smith, and B. Benhabib, "Three-dimensional location estimation of circular features for machine vision," *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 5, pp. 624–640, 1992.
5. L. Zhu and W. Weng, "Catadioptric stereo-vision system for the real-time monitoring of 3d behavior in aquatic animals," *Physiology & behavior*, vol. 91, no. 1, pp. 106–119, 2007.
6. R. Tillett, N. McFarlane, and J. Lines, "Estimating dimensions of free-swimming fish using 3d point distribution models," *Computer Vision and Image Understanding*, vol. 79, no. 1, pp. 123–141, 2000.
7. C. Spampinato, Y.-H. Chen-Burger, G. Nadarajan, and R. Fisher, *Detecting, Tracking and Counting Fish in Low Quality Unconstrained Underwater Videos*, 2008, vol. 2, pp. 514–519.
8. A. S. Kane, J. D. Salierno, G. T. Gipson, T. C. Molteno, and C. Hunter, "A video-based movement analysis system to quantify behavioral stress responses of fish," *Water Research*, vol. 38, no. 18, pp. 3993–4001, 2004.
9. G. Xiao, W. Zhang, Y.-L. Zhang, J.-J. Chen, S.-S. Huang, and L.-M. Zhu, "Online monitoring system of fish behavior," in *Control, Automation and Systems (ICCAS), 2011 11th International Conference on*. IEEE, 2011, pp. 1309–1312.
10. T. K. Kleinhappel, O. H. Burman, E. A. John, A. Wilkinson, and T. W. Pike, "Diet-mediated social networks in shoaling fish," *Behavioral Ecology*, vol. 25, no. 2, pp. 374–377, 2014.
11. M. Delafontaine, A. G. Cohn, and N. Van de Weghe, "Implementing a qualitative calculus to analyse moving point objects," *Expert Systems with Applications*, vol. 38, no. 5, pp. 5187–5196, 2011.
12. M. Hanheide, A. Peters, and N. Bellotto, "Analysis of human-robot spatial behaviour applying a qualitative trajectory calculus," in *RO-MAN, 2012 IEEE*. IEEE, 2012, pp. 689–694.
13. N. Mavridis, N. Bellotto, K. Iliopoulos, and N. Van de Weghe, "Qtc 3d: Extending the qualitative trajectory calculus to three dimensions," *Information Sciences*, vol. 322, pp. 20–30, 2015.
14. D. Carroll, E. Köse, and I. Sterling, "Improving frenet's frame using bishop's frame," *arXiv preprint arXiv:1311.5857*, 2013.
15. J. Heikkila and O. Silvén, "A four-step camera calibration procedure with implicit image correction," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. IEEE, 1997, pp. 1106–1112.
16. P. KaewTraKulPong and R. Bowden, "An improved adaptive background mixture model for real-time tracking with shadow detection," in *Video-Based Surveillance Systems*. Springer, 2002, pp. 135–144.
17. J. Canny, "A computational approach to edge detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 6, pp. 679–698, 1986.
18. A. Fitzgibbon, M. Pilu, and R. B. Fisher, "Direct least square fitting of ellipses," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 5, pp. 476–480, 1999.

Multipitch Estimation Applied to Single-Channel Audio Source Separation: Relevant Techniques and Challenges

Alejandro Delgado Castro and John E. Szymanski

Audio Lab, Department of Electronics, University of York
Heslington, York, North Yorkshire, YO10 5DD. United Kingdom
adc533@york.ac.uk
john.szymanski@york.ac.uk

Abstract. The estimation of melody trajectories in single-channel polyphonic signals is a major field of study in digital signal processing, with principal applications in audio source separation, automatic music transcription and musical genre identification. The YIN algorithm and one multiple fundamental frequency estimator are tested using two different mixtures of harmonic sounds in order to identify advantages and limitations. A strategy is presented as a way to overcome these limitations and hence improve the accuracy of the estimated pitch trajectories.

Keywords: Single-Channel Audio Source Separation, Fundamental Frequency Estimation, Multipitch Estimators, Principal Melody Extraction, Pitch Trajectories, YIN algorithm.

1 Introduction

In general, when most western musical instruments are excited to produce a note, when viewed in the frequency domain the result is not an isolated frequency but is rather a series of energy peaks having different frequencies and amplitudes. The spectral envelope characterized by these components is one of the cues that human brain uses to distinguish between different instruments, while the musical note itself can be characterized by its fundamental frequency or perceived pitch[1]. Within this harmonic distribution of energy, the fundamental frequency works as the reference point for generating the rest of the harmonically related components of the note. However, the fundamental frequency does not always coincide with the highest peak in magnitude.

The number of fundamental frequencies present in a mixture of sounds is probably one of the most important parameters that can be used to estimate the number of sources, and it can also be used to further characterize those sources individually. Melody trajectories are usually generated when the pitch

[1] Despite the fact that *Fundamental Frequency* is usually considered as feature of the signal, while *Pitch* refers more to a perceptual measure, the terms will be used interchangeable across the article.

of one or more identified sources is tracked across time. In order to implement an accurate source separation system for harmonic sounds, a reliable fundamental frequency estimator has to be used to generate a reliable pitch trajectory for every source.

Many fundamental frequency estimation methods have been presented [2, 4, 14], both for monophonic and polyphonic signals, and some of them have proven to be an important preprocessing stage for audio source separation algorithms, in particular, for those methods concentrated in isolating or extracting harmonic sounds from single-channel music recordings.

The aim of this article is to give a general review of some established fundamental frequency estimators, namely the YIN algorithm [2] and one multipitch estimator by Duan et. al [4], focussing on their principal features and limitations. Several tests are presented here, using individual and mixed sounds, in order to reveal interesting characteristics and evaluate their performance. The contribution of this paper is to present some proposals on how to improve the accuracy of these algorithms.

2 Fundamental Frequency Estimation in Monophonic Signals

The problem of estimating fundamental frequencies in audio signals was first studied in monophonic recordings and was then applied to speech processing [11]. Since then, many other methods emerged and were specifically designed for music signals. The vast majority of existing algorithms divide the entire time-domain representation of the input signal into short portions, called frames, and then present fundamental frequency estimates for every frame. The so-generated sequence of pitches can be considered as an appropriate representation of the melody trajectory [5].

2.1 Classification

Pitch estimators can be grouped into several types according to the main principle or function that is used to approximate the set of fundamental frequencies. The most common types are listed and explained briefly below.

- **Zero-Crossing Rate.** This is probably the simplest and most inexpensive type of pitch estimator, and consists of counting the number of times the input signal crosses the reference axis (zero level) in order to detect periodicity. Although the method is simple, its results are unreliable when applied to noisy signals. It also struggles to deliver accurate results for those harmonic signals where the fundamental partial is not the strongest.
- **Autocorrelation.** Some of the most frequently used algorithms in pitch detection are based on autocorrelation functions. Periodicity in this case is indicated by the maximum of this function. Therefore, autocorrelation methods select the highest non-zero lag to compute the estimated period [14].

Algorithms in this category have proven to be relatively robust against issues caused by noise, but sensitive to particularities in the spectral characteristics of sounds [5].

- **Cepstrum Analysis.** Cepstrum-based pitch detectors were the first methods to be realizable through digital computation and were used as reference for other algorithms [5]. The cepstrum of the input signal is the inverse Fourier transform of the logarithm of its power spectrum. A peak-picking schema is used to find strong peaks within this function which indicates the underlying periodicity. Cepstral methods normally perform poorly in noisy environments and frequent octave errors have also been reported. However, good results have been obtained when dealing with formants in speech processing [5].
- **Harmonic Matching Methods.** This type of algorithm identifies a fundamental period by analysing a pattern of spectral peaks in the magnitude spectrum. When all the relevant peaks are identified, the most likely fundamental frequency is found that is consistent with the pattern of the observed peaks. Some related drawbacks usually emerge when the levels of noise are high, or when the deviation between ideal harmonics and real partials is also high.
- **Wavelet-based Algorithms.** Multiresolution and multi-scale analysis are techniques that have been also applied to pitch detection. In contrast to Fourier Analysis, the Wavelet Transform utilizes different resolutions to analyse high and low frequency regions. Hence, it can be considered as a form of constant Q frequency analysis.

2.2 The YIN Algorithm

One method that has been widely used in several application areas of pitch detection, is the so-called YIN Algorithm, developed by De Cheveigné et. al. [2]. It is based on the autocorrelation function and incorporates a number of modifications that are combined to prevent errors. According to [2], the most relevant features of the YIN algorithm are presented as follows.

- The error rates were reported to be three times lower than other competing methods.
- It can be applied to either speech or music signals.
- It is suitable for high-pitched voices or music since there is no upper limit on the frequency search range.
- The algorithm has a small number of parameters and any fine tuning of them is not required.

When the YIN algorithm is applied to a non-stationary monophonic signal, the result is a series of frequency estimates that can be used to construct the melody trajectory of the sound. Figure 1 shows the estimated melody trajectory of three different audio signals: a short excerpt of an aria for soprano, a modern viola playing the note A4 without vibrato, and a clarinet playing five different

musical notes. The original audio recordings were taken from the Open Air Anechoic Audio Database [9].

The depicted results show that the estimation of fundamental frequencies is highly accurate during the sustain of every note. However, some problems arise during note transitions or silences, where the algorithm produces spurious peaks of short duration. Considering the pitch trajectory for the soprano voice, the spurious peaks can be spotted easily, and they correspond to short silences between different notes. For the clarinet sound, the first and fourth transitions are also problematic in this sense. On the other hand, it can also be observed that the algorithm was able to track the vibrato in those sung notes, which is advantageous if the melody trajectory is going to be used for source separation.

If the YIN algorithm is applied to a mixture of different sounds, i.e. a polyphonic signal, additional problems will appear and the estimated trajectory will not always be accurate. The reason for this failure is that YIN does not perform multipitch estimation, so the algorithm always assumes that only one source is present in every frame. Multipitch estimators will be introduced in the following section.

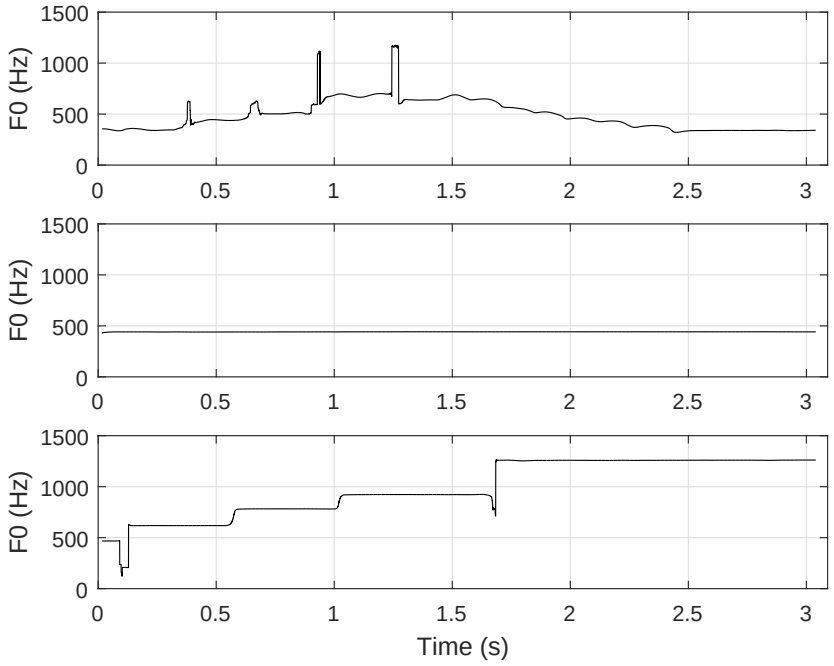


Fig. 1. Estimated melody trajectories for different audio signals, using YIN Algorithm [2]. Singing voice (Upper Chart), Modern viola playing the note A4 (Middle Chart), Clarinet playing the notes A♯4, D♯5, G5, A♯5, and D♯6 (Lower Chart).

3 Fundamental Frequency Estimation for Polyphonic Signals

Multiple fundamental frequency estimation algorithms assume that the input signal $x(t)$ is a mixture of two or more harmonic sources $\tilde{x}(t)$ plus a residual non-harmonic signal $z(t)$. Hence, the model imposed on the input signal can be expressed by the following equation, after [14].

$$x(t) = \tilde{x}(t) + z(t) \approx \sum_{m=1}^{M} \sum_{h=1}^{H_m} A_{m,h} \cos(h\omega_m t + \phi_{m,h}) + z(t) \tag{1}$$

Where H_m is the total number of harmonics to be considered, M is the number of sources and $A_{m,h}$ is the amplitude of the sinusoidal component associated with the *h-th* harmonic of the *m-th* source. In this way, the multipitch estimation problem consists of estimating the number of sources present and their fundamental frequencies [14]. The way in which the multipitch estimator handles overlapping harmonics, transients and reverberation, significantly improves or degrades the accuracy of the results.

3.1 Brief Review of Multipitch Estimation Algorithms

Many multipitch estimation algorithms are used as preprocessing stages for audio source separation. Some of those most commonly used are discussed below.

In 2003, Klapuri presented an iterative method for multiple fundamental frequency estimation based on bandpass filtering and spectral cancellation [7]. The pitch associated with the most prominent sound was estimated first and then its harmonic structure was subtracted from the original mixture, via a spectral smoothness principle. The cycle was repeated using the residual spectrum in order to extract a second pitch and continued until the estimated number of sources in the mixture was reached.

In 2008, a refined approach by Klapuri was presented [8], based on the human auditory system. The model upon which the method was designed can be described as a filter bank that decomposes the original signal into a fixed number of subbands. To achieve such a decomposition, a Gammatone filter was used. The obtained results were reported to be satisfactory when compared to other competing algorithms. The proposed system inspired parallel approaches in source separation techniques, such as [12], where multipitch estimation was used to guide a set of spectral filters to extract harmonic sources from single-channel recordings.

Other methods have concentrated on using probabilistic approaches as a way to select the fundamental frequencies that better explain a given distribution of partials. The method proposed in [4] models the original spectrum as the combination of two regions: spectral peaks and non-peaks. The signal is broken into frames and the maximum-likelihood approach is used to estimate the pitches of the detected harmonic sources. A polyphony estimation method and additional

refinement stages were also presented. This particular algorithm will be explored further in Section 3.2.

Bayesian harmonic models were also used in [13] for separating harmonic sources in single-channel recordings. Multipitch estimation was used as a pre-processing stage, and it was carried out in two stages. During the first stage, all fundamental frequencies without octave relations were estimated. Then, the undetected pitches were resolved within the second stage, based on continuity of the magnitudes of harmonic partials.

A different approach was presented in [10] where the Continuous Complex Wavelet Transform (CCWT) was used as a time-frequency representation of the mixture. A Morlet wavelet was used as a filter bank to decompose the original signal into several sets of wavelet coefficients, and a novel type of scalogram was built from the data. According to a set of rules, the most relevant peaks were chosen and used to estimate the fundamental frequency candidates.

The regularized least-squares was used in [6] as a solution for simultaneous sparse source selection and parameter estimation. By exploring the block sparsity, the algorithm allows the estimation of fundamental frequencies to track a set of identified sources, without *a priori* assumptions of the number of harmonics for each source. The addition of a Bayesian prior probability distribution and regularization coefficients was considered, in order to efficiently incorporate both earlier and future blocks in the tracking of frequency estimates.

3.2 Exploring a Multipitch Estimator: A Case Study

The multipitch estimator proposed in [4] has been tested using two types of audio mixtures. First, a combination of two sounds produced by harmonic instruments, and then second, some background voices and drums were also incorporated into the mixture. Here, results are presented and discussed. Moreover, they are the basis for future possible improvements that are proposed in the following section.

The algorithm by Duan et. al. [4] was previously introduced as a probabilistic approach for fundamental frequency estimation. An implementation of this algorithm in Matlab is available for research from the author's webpage [3]. The supplied code was used during the tests in order to evaluate its accuracy and reliability as a melody extraction system.

The first test applied Duan's multipitch estimator to a mixture of two harmonic sounds. Melody trajectories for these two sounds were individually obtained in previous sections, using the YIN algorithm [2], and presented in Figure 1 (the last two plots). To compare the results of Duan's multipitch estimator, the YIN algorithm [2] was also applied to the audio mixture. Final results for the first test are presented in Figure 2.

Observing the output trajectory generated by YIN, it can be inferred that the method was unable to completely estimate the pitch associated with the modern viola. Also, YIN did not detect the pitch trajectory of the clarinet. The Duan multipitch estimator [3][4], on the other hand, correctly identified the two sources and their pitch trajectories.

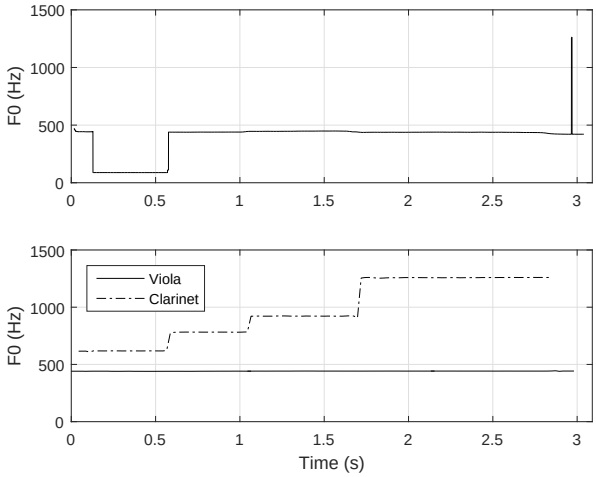


Fig. 2. Pitch trajectories extracted from a polyphonic audio signal comprising two musical instruments: Modern Viola and Clarinet. Using YIN Algorithm [2] (Upper Chart). Using Duan's Multipitch Estimator [3][4] (Lower Chart).

At this point, the good performance of the multipitch estimator is not really surprising, because the mixture was a simple mix of two individual harmonic sounds, and their pitches were selected so they were not in an octave relation. Furthermore, considering that both instruments were recorded under anechoic conditions, the levels of reverberation and noise are negligible. Also, the non-existence of additional background instrumentation reduces significantly the risk of delivering misleading results.

In order to test the algorithm under more realistic conditions, a second test was conducted. The same mixture of two harmonic sounds was combined with background instrumentation consisting of voices and drums. These additional sounds were taken from the Sixth Community-Based Signal Separation Evaluation Campaign 2015 database [1]. Multipitch estimation, using Duan's algorithm [3][4], was applied to the resulting mixture and the extracted pitch trajectories are shown in Figure 3.

The new estimated melody trajectories indicate that the multipitch estimator produced misleading estimates for some specific time-domain frames. There are sections where the algorithm swapped the correct frequency estimates between the two sources. Further, in other frames, the estimator was unable to detect correctly the fundamental frequencies of the second instrument.

Unfortunately, most commercial recordings also have significant levels of background instrumentation and noise, so that, overcoming such issues is an essential step in enlarging the range of audio recordings that can be processed, and improving the quality of the estimated sources.

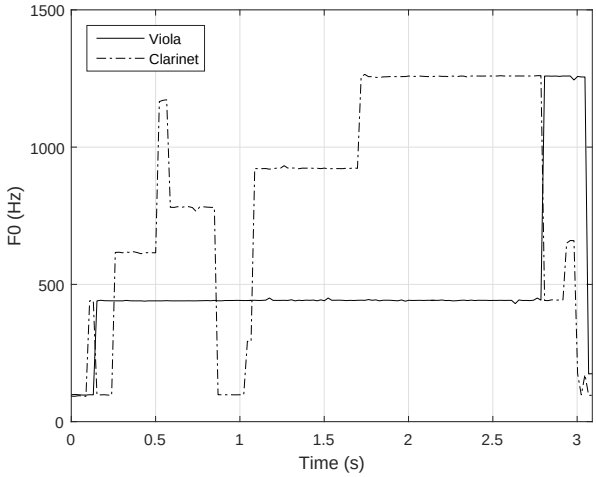


Fig. 3. Pitch trajectories extracted from a polyphonic audio signal comprising two musical instruments plus background voices and drums, using Duan's algorithm [3][4].

4 Challenges and Future Improvements

A possible way to overcome some of the limitations described in the previous section is by introducing additional stages into the system to assist the multipitch estimation algorithm in observing the input signal in an effective way. The rationale for this is that, *a priori*, there is no knowledge regarding the signal content, so that the existing algorithms can be misled by volume differences between different harmonic and non-harmonic sources. Hence, in theory, it is necessary to have more than one observation of the same input mixture.

Since multiple observations are not available the only way to provide additional input is to produce modified versions of the original signal. This should help in selecting the most reliable frequency estimates for every pitch trajectory. The stages of the proposed strategy are explained below.

- The original mixed signal is passed through a set of digital high-pass filters, each one having a different cutoff frequency. As a result, several different versions of the original audio mixture are generated.
- Multipitch estimation is applied to every filtered version of the input signal in order to estimate a set of pitch trajectory candidates.
- These trajectories are arranged in a data structure that has the form of a matrix in which the columns follow the frame number and the rows correspond to the filter used to generate that particular pitch trajectory.
- To evaluate the fundamental frequency estimates in every frame, a measurement of salience is proposed. The process assumes that those frequencies associated with a real pitch trajectory are supposed to exhibit a clear and

structured harmonic pattern, while spurious non-harmonic frequency estimates are not suppose to have any structured distribution in the magnitude spectrum. Therefore, the salience can be measure by calculating the energy of the first few partials associated with every frequency estimate in the frame. These result can be used to organize the fundamental frequency candidates, and those having the highest energy content are considered as the most reliable.
- A continuity-based approach can be also considered for error correction during melody trajectory estimation. If the multipitch estimator is using a short frame, it can be assumed that a normal note has to be present across several adjacent frames. If some unusually rapid change in pitch occurs and it is shorter than the minimum expected note duration, the related frames can be labelled as misleading and replaced with values from the data structure, that preserve the continuity of that particular melody trajectory.

This iterative structure, in which multipitch estimation is applied to different versions of the input audio mixture, is a promising alternative to extract pitch trajectories for harmonic sources in those cases where significant levels of background instrumentation or noise are present.

The proposed strategy requires control parameters to be defined, for example, the number of filters that will be used and their corresponding cutoff frequencies. Also important are the number of partials that will be considered for the salience measurement and the minimum accepted duration for musical notes. Assigning adequate values for these parameters will require further research and tests. Hence, establishing the impact of these parameters on the overall performance of the system is the essential next step in the work.

5 Conclusions

Multipitch estimation in single-channel recordings has been addressed and some relevant algorithms were described and tested using different types of audio inputs. The YIN algorithm and one multipitch estimator by Duan were evaluated for robustness in principal melody extraction tasks, using recordings of harmonic instruments, background voices and drums.

The results obtained confirmed the stability of the YIN algorithm as a melody extraction system for individual harmonic sounds, while Duan's multipitch estimator outperformed the YIN algorithm and delivered positive results for mixtures of two different harmonic sounds with unrelated fundamental frequencies. When background voices and drums were added to the mixture, the estimated melody trajectories showed swapping errors in some frames, while other sections were misleadingly estimated.

A strategy was proposed as a possible way to overcome these limitations and improve melody extraction systems for polyphonic signals. The incorporation of a set of high-pass digital filters, energy-based evaluation of pitch candidates, and continuity, are aspects that will be investigated as possible ways of producing

more reliable multipitch estimation for audio source separation systems and many other applications.

Acknowledgements

The authors would like to thank the University of Costa Rica (UCR), and the Costa Rican Ministry of Science, Technology and Telecommunications (MICITT) for their support in funding this research.

References

1. Sixth Community-Based Signal Separation Evaluation Campaign, https://sisec.inria.fr/sisec-2015/2015-underdetermined-speech-and-music-mixtures/
2. De Cheveigné, A., Kawahara, H.: YIN: A Fundamental Frequency Estimator for Speech and Music. The Journal of the Acoustical Society of America 111(4), 1917–1930 (2002)
3. Duan, Z.: Multi-Pitch Analysis, http://www.ece.rochester.edu/~zduan/multipitch/multipitch.html
4. Duan, Z., Pardo, B., Zhang, C.: Multiple Fundamental Frequency Estimation by Modeling Spectral Peaks and Non-Peak Regions. IEEE Transactions on Audio, Speech and Language Processing 18(8), 2121–2133 (2010)
5. Gomez, E., Klapuri, A., Meudic, B.: Melody Description and Extraction in the Context of Music Content Processing. Journal of New Music Research (2003)
6. Karimian Azari, S., Jakobsson, A., Jensen, J.R., Christensen, M.G.: Multi-Pitch Estimation and Tracking Using Bayesian inference in Block Sparsity. In: 23rd European Signal Processing Conference (EUSIPCO). pp. 16–20. No. 2, IEEE (2015)
7. Klapuri, A.: Multiple Fundamental Frequency Estimation Based on Harmonicity and Spectral Smoothness. IEEE Transactions on Speech and Audio Processing 11(6), 804–816 (2003)
8. Klapuri, A.: Multipitch Analysis of Polyphonic Music and Speech Signals Using an Auditory Model. IEEE Transactions on Audio, Speech, and Language Processing 16(2), 255–266 (2008)
9. Murphy, D., Shelley, S.: Open Air Anechoic Audio Database, http://www.openairlib.net/
10. Ponce de Leon Vazquez, J., Beltrán, F., Beltrán, J.R.: A Complex Wavelet Based Fundamental Frequency Estimator in Single-Channel Polyphonic Signals. Proc. Digital Audio Effects (3), 1–8 (2013)
11. Salamon, J., Gomez, E., Ellis, D.P.W., Richard, G.: Melody Extraction from Polyphonic Music Signals: Approaches, Applications, and Challenges. IEEE Signal Processing Magazine 31(February), 118–134 (2014)
12. Siamantas, G.: An Iterative, Residual-Based Approach to Unsupervised Musical Source Separation in Single-Channel Mixtures. Phd, University of York (2009)
13. Wang, Y., Wang, H., Zhu, B., Wang, X.: Single-Channel Polyphonic Signal Separation Based on a Novel Multi-F0 Estimation Method. In: 14th International Conference on Communication Technology. pp. 1334–1338. IEEE (2012)
14. Yeh, C.: Multiple Fundamental Frequency Estimation. Phd, University of Paris VI (2008)

Towards a Confidence-Centric Classification Based on Gaussian Models and Bayesian Principles

Dongxu Han, Hongbo Du and Sabah Jassim

Department of Applied Computing, University of Buckingham, Buckingham MK18 1EG, UK
{dongxu.han, hongbo.du, sabah.jassim}@buckingham.ac.uk

Abstract. Classification is an essential form of machine learning. It has been widely used in various application areas. Conventional classification schemes are mostly interested in the class label outcome rather than the strength of the class predictions. This paper introduced a new interpretation of "classification confidence" to complement the predicted class label. The level of confidence is formulated based on Gaussian probability models and the Bayesian classification principles. The paper shows the soundness of the concept and argues the very essence of embedding it in the centre of a classification process. The paper applies the proposed concept over a data set of miscarriage cases in early pregnancy and demonstrates that the concept works well and indeed reflects the degree of certainty regarding the data set.

Keywords: confidence, classification, Gaussian probability models, Bayesian principles

1 Introduction

Classification is one of the fundamental processes in machine learning. It has been widely applied in various applications. In a typical classification process, a model is trained by applying a learning algorithm to a set of training examples. The trained model is then applied to an unseen data sample to determine its class label. Often, a trained model is also tested on a set of testing examples to determine the level of accuracy of the model [1]. Various types of classification models and learning algorithms have been developed over the past several decades [2].

In recent years, classification techniques are increasingly deployed in medical diagnosis with promising results. Classification models are trained on various data features extracted from medical images and observations from medical tests. The models are then used in clinical examinations as a complementary tool for more accurate and timely diagnoses of diseases. In this type of real-life application scenarios, predicting the correct class label alone may not be sufficient since the classification decision must have adequate support especially when there are significant risky implications of misclassification. For example, in a scenario of diagnosing the right type of a tumour, it is important for doctors to know whether the tumour is benign or malignant *and* how much belief they have in that diagnostic decision. Such a requirement may also apply

to many other application domains beyond medical diagnosis. Unfortunately, conventional approaches in classification are more interested in the predicted class label rather than how reliable the prediction is. The tested level of accuracy for a model can only provide a rough idea of the model's general reliability, but does not give indication about the level of certainty in each specific classification decision.

Various forms of similar concepts about classification confidence have been considered in the past. A k-nearest neighbour model may use the proportion of the majority in a majority voting scheme to indicate the level of confidence [3]. Decision tree classifiers can use the proportion of the majority of the training examples at a leaf node to determine the level of certainty [4]. But the preciseness of this type of measurement for confidence evaluation could be very much limited. A Support-Vector-Machine classifier may quantify its confidence by calculating the distance from the target data point to the decision hyper-plane [5], but this logic may have a limited range of application. In a recent work involving the second and the third authors [6], a simple criterion of confidence was developed. The scheme categorises each decision with a confidence band of either high, medium or low, based on thresholds that are set according to the boundaries between the classes, but these discrete bands are also limited. A similar but more formal approach to address this issue comes from conformal predictions [7]. In this approach, an error probability ε is introduced when a classifier is making decisions, and then the level of confidence can be seen as $1 - \varepsilon$, i.e. the probability of correct classifications. It provides a very nice regression over discrete data samples, but modelling of the error rate ε by most likely following a Bernoulli distribution still seems quite naïve [8] [9].

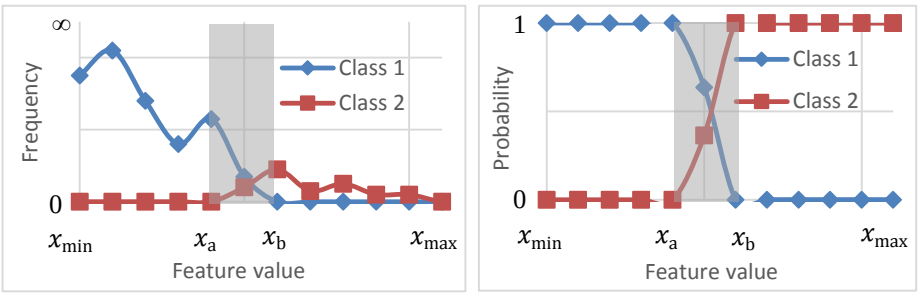
In this paper, we propose a new approach to address the issue of classification confidence from a slightly different angle of view than that in conformal predictions. The proposed approach defines the level of confidence as the difference between the posterior probability of predicting the class and the posterior probability of predicting other classes. The rationale behind this is quite straightforward: when the difference is large, the posterior probability of predicting one class is high, and hence the level of confidence is high for that class. When the difference is small, the posterior probabilities of predicting the possible classes are near to each other and hence the level of confidence about one of the classes should be low. In an extreme situation of equal probabilities, the level of confidence should be zero. The proposed approach is based on Gaussian models, and the definition of classification confidence is based on the Bayesian theorem. We shall present the rationale behind the concept and test its application by conducting an empirical study over a set of early pregnancy data. The evidence shows the validity of the concept as well as its practical potentials. We believe that the concept can be adopted not only with the outcome from one classifier but also in a confidence-based fusion framework in a complex decision support system.

The rest of the paper is organised as follows. Section 2 presents our definition of the classification confidence and the classification modelling in light of the confidence. Section 3 shows the experimental work of using the concept for quantifying prediction strength of miscarriage cases in early pregnancies. Section 4 discusses some related issues regarding the level of confidence. Section 5 concludes the paper with a summary of the work reported and future work needed.

2 The Proposed Method

2.1 Classification Confidence

In a typical training data set, examples of the individual classes may be distributed differently in the corresponding dimensional space. Figure 1 presents a simplified view of distributions of a set of one-dimensional training examples of two classes, and provides a conspicuous view about the strength of classification for each class. As illustrated by the frequency diagram in Figure 1(a), the two classes are very much distinct from each other when the data feature x has a value that is below a certain threshold x_a or above another threshold x_b due to the lack of examples from the opponent classes. However, conflicts of classification occur in a region between the two thresholds, where samples' feature of two classes starts to overlap. At the intersection point of the two curves, the overlapping occurs the most. Therefore, the overlapped region should be considered as the "zone of confusion" and the level of uncertainty in classifying sample attains the maximum value when the presences of the two classes are nearly equal.



(a) Frequency Distribution of Feature Values (b) Probability of a Class over Feature Values

Fig. 1. An Illustration of Value Distributions of Examples of 2 Classes

Based on this understanding, it is logical to transfer the previous frequency-based reasoning into a probability-based concept as shown in Figure 1(b), where the likelihood of the presence of different classes is a good indication of the confusion caused in classification. As illustrated in the diagram, all the discussed characteristics regarding "confusions" are well preserved with a normalised scale. The range between the two probability curves on the y-axis indicates the magnitude of the overlapping between the two classes, which should be considered as being proportional to the level of decision confidence.

Therefore, for a given finite class set $\{\omega\} = \{\omega_1, \omega_2, \dots, \omega_k\}$, the level of confidence can be presented as:

$$\text{Confidence}(\omega_i|\vec{x}) \propto |\text{P}(\omega_i|\vec{x}) - (1 - \text{P}(\omega_i|\vec{x}))| \quad (1)$$

where $\text{P}(\omega_i|\vec{x})$ is the conditional probability of predicting class ω_i based on a given feature vector $\vec{x}$ and therefore the aggregate probability of predicting into the rest of the classes will be $1 - \text{P}(\omega_i|\vec{x})$.

The second term in the absolute difference in (1) is indeed the classification error rate ε for class ω_i at the given data point, which can be simplified as:

$$\text{Confidence}(\omega_i|\vec{x}) = |2\text{P}(\omega_i|\vec{x}) - 1| \tag{2}$$

This definition is justified by an assumption that the level of the confidence of the classification is directly proportional to the difference of the two probabilities without any transition bias, i.e. the gradient is equal to 1. Expression (2) motivates the introduction of a generalised confidence-centric score function for the classified label ω_i as:

$$\text{Decision score}(\omega_i|\vec{x}) = 2\text{P}(\omega_i|\vec{x}) - 1 \tag{3}$$

Here the sign of the decision score indicates the belonging of the class, which positive value would indicate a confirmation of the chosen class ω_i and negative value indicates a preference of the other classes. The absolute value of the decision score is the level of confidence in the decision made on the class belongings.

2.2 Decision Score Modelling

Here we introduce a Bayesian-based model for the decision score defined in equation (3). According to the Bayesian theorem:

$$\text{P}(\omega_i|\vec{x}) = \frac{\text{P}(\vec{x}|\omega_i)\text{P}(\omega_i)}{\text{P}(\vec{x})} \tag{4}$$

where $\text{P}(\omega_i)$ and $\text{P}(\vec{x})$ are two priors that represent the natural incidence of the class ω_i and the expected observation probability of feature $\vec{x}$, while $\text{P}(\vec{x}|\omega_i)$ is known as a posterior of the feature $\vec{x}$ given that it belongs to the class ω_i. Unfortunately, it is impossible to know exactly the priors in real-life scenarios due to unavoidable uncertainty and randomness. We, therefore, estimate the parameters by using the training dataset.

Given a sample space $\Omega = \{[\omega_1],[\omega_2],...,[\omega_k]\}$, where $[\omega_i]$ is the set of all samples that belong to class ω_i, then $\text{P}(\omega_i)$ can be estimated as the proportion of the interested class ω_i to the total number of samples i.e.:

$$\text{P}(\omega_i) = \frac{|[\omega_i]|}{|\Omega|} \tag{5}$$

$\text{P}(\vec{x})$ and $\text{P}(\vec{x}|\omega_i)$ are the two probability functions describing the distribution of the feature $\vec{x}$, respectively within the overall population and within the population of class ω_i. Our proposed scheme assumes that both are Gaussian distributions. Consequently, a simplified model that is based on a single Gaussian distribution is proposed first as follows. Given the mean μ and variance σ^2 for a univariate feature $\vec{x}$, we use the Gaussian probability density function:

$$\mathcal{N}(x \mid \mu, \sigma^2) = \frac{1}{\sqrt{2\sigma^2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

for characterizing $\text{P}(\vec{x})$ and $\text{P}(\vec{x}|\omega_i)$ as:

$$\begin{cases} \mathrm{P}(\vec{x}) = \mathcal{N}(x \mid \mu_{\Omega}, {\sigma_{\Omega}}^2) \\ \mathrm{P}(\vec{x}|\omega_i) = \mathcal{N}(x \mid \mu_{\omega_i}, {\sigma_{\omega_i}}^2) \end{cases} \tag{6}$$

In many applications, data features normally exist in a multidimensional space. Therefore, it is essential that we expand the previous simple model into a multivariate Gaussian model to accommodate multidimensional feature vectors. So for a given *d* dimensional data set with the mean vector $\vec{\mu}$ and covariance matrix Σ, we simplify the standard Gaussian probability density function $\mathcal{N}(\vec{x} \mid \vec{\mu}, \Sigma)$ as:

$$\mathcal{N}(\vec{x} \mid \vec{\mu}, \Sigma) = \frac{1}{\sqrt{2\pi^d |\Sigma|}} e^{-\frac{(\vec{x}-\vec{\mu})\,\Sigma^{-1}\,(\vec{x}-\vec{\mu})^T}{2}}$$

We then derive $\vec{\mu}_x, \Sigma_x$ from $\omega = \{\omega_1, \omega_2, \ldots, \omega_k\}$ and $\vec{\mu}_{\omega_i}, \Sigma_{\omega_i}$ from ω_i, and then $\mathrm{P}(x)$ and $\mathrm{P}(x|\omega_i)$ can then be characterized as:

$$\begin{cases} \mathrm{P}(\vec{x}) = \mathcal{N}(\vec{x} \mid \vec{\mu}_{\Omega}, \Sigma_{\Omega}) \\ \mathrm{P}(\vec{x}|\omega_i) = \mathcal{N}(\vec{x} \mid \vec{\mu}_{\omega_i}, \Sigma_{\omega_i}) \end{cases} \tag{7}$$

One concern is that we have tried only to use a single Gaussian in the modelling so far, but this may not always be realistic. Real-life data may reflect a combination of multiple Gaussians, each of which has its own mean vector and covariance matrix. Therefore, we have chosen to further extend our model into a Gaussian Mixture Model (GMM). In the mixture model, each sub-Gaussian model has been given a parameter set $\theta = \{W, \vec{\mu}, \Sigma\}$, where *W* represents the weight of each sub-model in the mixture and the summation of the weight of all the models should be 1. Therefore, given a sequence of *K* parameter sets $\{\theta_{i=1\ldots K}\}$, i.e., *K* Gaussian sub models, we can characterize our universal mixture model as:

$$\mathcal{N}(\vec{x} \mid \theta_{i=1\ldots K}) = \sum_{i=1}^{K} W_i \mathcal{N}(\vec{x} \mid \vec{\mu}_i, \Sigma_i)$$

Therefore, we would be able to derive parameter set θ_{ω_i} for each class from the relevant class set $\{\omega_i\}$ and the weight of each set would be considered as its proportion in the whole training set, i.e., $W_i = \frac{|[\omega_i]|}{|\Omega|}$, which $\mathrm{P}(\vec{x})$ and $\mathrm{P}(\omega_1)\mathrm{P}(\vec{x}|\omega_i)$ can then be characterised as:

$$\begin{cases} \mathrm{P}(\vec{x}) = \mathcal{N}(\vec{x} \mid \theta_{\omega_{i=1\ldots k}}) \\ \mathrm{P}(\omega_i)\mathrm{P}(\vec{x}|\omega_i) = \mathcal{N}(\vec{x} \mid \theta_{\omega_i}) \end{cases} \tag{8}$$

3 Experiment Results

To evaluate the usefulness of the confidence concept introduced in the previous section, this study conducted several experiments using a collection of data about early pregnancies obtained from the Early Pregnancy Department, Queen Charlotte and

Chelsea Hospital, Imperial College London. The data collection contains three measurements of gestational sac sizes, i.e. major and minor diameters from the sagittal plane and major diameter from the transverse plane, recorded from the ultrasound machine. The collection also includes a variable, known as the Mean Sac Diameter (MSD), derived from the three diameter measurements. The collection consists of a training set of 94 examples (15 cases of miscarriage (MC) and 79 cases of PUV (Pregnancy of Unknown Viability)), and a test set of 90 examples (11 cases of MC and 79 cases of PUV)). Basically, PUV is declared when there are no clear signs of miscarriage although this may occur in subsequent scans.

We first trained the proposed decision score model and derived their parameters based on the training dataset. We then applied the proposed models on each testing example from the test set and measured the decision score for each testing example. As introduced in the proposed method section, the decision score would take a range of [-1, 1], which can be seen as the decision confidence towards [PUV, MC] in this binary class dataset.

3.1 Evaluation of the "Zone of Confidence"

Figure 2 presents scatterplots of the decision scores against feature values along the MSD dimension. According to the known literature in the related field of medicine, 25mm in MSD is a well-recognised threshold for separating PUV from miscarriage cases [10]. Therefore, we have rescaled the MSD dimension by setting (25, 0) as the origin, then plotted the related decision score of each feature value in the test set for each model accordingly. The corresponding class provided in the test set was plotted in markers of triangle and cross respectively. After the rescaling, the 1st, 2nd, 3rd and 4th quadrants in each scatterplot would indicate the possible classification results, i.e. true positive, false positive, true negative and false negative respectively. The figure shows that not only the confidence scores are very high for MC (≥31mm) and PUV (≤16mm), but also the existence of a confusion zone between 16mm and 31mm with the maximum confusion near the threshold of 25mm. This finding itself is interesting due to a well-known fact that 16mm was a previous threshold in use that is only recently being revived to 25mm because of concerns of potential false positives from some doctors [10]. This finding indicates that the confidence score does reflect the level of confidence in the diagnosis.

In addition, all the modelled data points were distributed according to a sigmoid pattern, which matches our expectation that the confidence would drop dramatically when it approaches the confusion point, i.e. the origin in the presented diagrams, or otherwise be stable at -1 or 1 when the feature value is outside the "confusion zone". The scatterplot also shows that the use of GMM has resulted in data confusions being moved from the false negative region into the false positive region.

Figure 3 shows the scatter plots of decision scores and feature values for multivariate situations. To clearly demonstrate the relationship between the decision scores and feature vector values, we on purpose combined the 3 diameter components of each feature vector into a single average value (in fact MSD), and display the location of the

data point along the MSD dimension. At the same time, the decision scores are calculated using the original 3D feature vectors themselves.

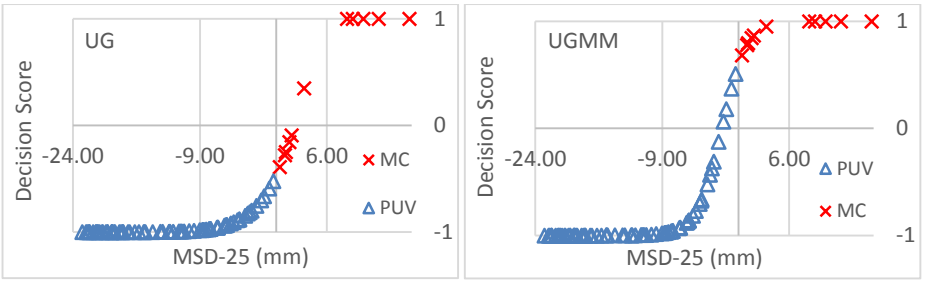


Fig. 2. Scatterplots of the MSD feature vs. decision scores in UG and UGMM situations (**UG: Univariate Single Gaussian on MSD; UGMM: Univariate Gaussian Mixture Model on MSD)*

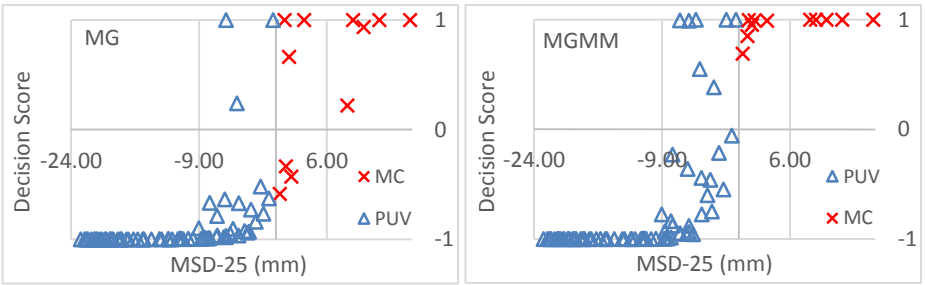


Fig. 3. Scatter plot of the feature values vs. decision scores in MG and MGMM situations (**MG: Multivariate Single Gaussian on 3 diameters; MGMM: Multivariate Gaussian Mixture Model on 3 diameters)*

Some general observations can be made from the scatterplots in both figure 2 and figure 3. The confusion zone clearly exists between the two thresholds, and the best fitting curve through the confusion zone tends to be close to a sigmoid line. The use of GMM also tends to move confusion cases from the false negative region into the false positive region, and at the same time increase the level of classification confidence in the true positive region. However, the confidence scores are more scattered in the confusion zone here than the scores for a univariate situation. The second scatterplot shows increased cases of misclassification even with high level of confidence.

In summary, the single Gaussian models tend to have a smoother fit to the sigmoid function, which reflected the nature of the proportional relationship between the MSD and the classification result. However, the performance of the multivariate Gaussian models shows that the data points are eventually made more distinguishable and pushed the classification results towards the two extremes, which provides a good sign for multivariate fusion in differentiating classes in highly overlapped feature values.

3.2 Evaluation of the Decision Score

As demonstrated in the previous section, the proposed method provides a good indication of the range of confidence/confusion. This study, however, also intends to evaluate the closeness of the decision score to a level of confidence exercised by manual diagnosis. The provided labels in the test set were the diagnosis results made by field experts based on known research truth.

Unfortunately, such a "manual diagnosis confidence" is not readily available in the data set provided. We, therefore, converted each of the given labels into to a decision score by mapping PUV and MC into -1 and 1 respectively, i.e. it is assumed that each of the decisions made by the experts was with absolute confidence. Following this, we would be able to evaluate the experiment result by calculating the difference between the human decision score and the derived decision score from the proposed confidence model. This difference would take a range of [0, 2], which 0 indicates a perfect match, and 2 would indicate an absolute conflict between the two decision scores. This evaluation was applied to each test example, and the average has been shown as an average error margin in Table 1.

Table 1. Confidence Error Margin Against Human Decision

Error Margin	*UG*	*UGMM*	*MG*	*MGMM*
MC	0.620	0.098	0.503	0.048
PUV	0.047	0.138	0.110	0.247
All	0.117	0.134	0.158	0.222

*UG: Univariate Single Gaussian on MSD; MG: Multivariate Single Gaussian on 3 diameters; UGMM: Univariate Gaussian Mixture Model on MSD; MGMM: Multivariate Gaussian Mixture Model on 3 diameters

The experiment result shows that SG best reflected the confidence nature of the PUV cases whereas MGMM had a better reflection on MC cases, which were only 0.047 and 0.048 units away from the human perspective on average. The experts had considerably more confidence on classifying MC cases base on well-understood diagnostic threshold [10]. Nevertheless, we have noticed that both SG and MG performed badly on fitting human perspectives on MC cases, which were 0.62 and 0.503 units less confident on average. It could be well caused by the domination of the PUV cases in the training data set, which makes the modelled natural expectation $\mathrm{P}(x)$ being very close to the probability of PUV classes.

This error can be well resolved by adopting GMM, which reduced the error margin of the SG and the MG to 0.098 and 0.048 respectively. However, the use of GMM may increase the error margin of the PUV cases at the same time. Nevertheless, it can still be tolerated since the influence was small, and after all, we should not expect the human experts being extremely confident on classifying PUV cases.

In general, the SG seems best fitting the human decisions. This is understandable since the real diagnoses made by the radiologists were based on the MSD measurement,

which makes the model that use univariate features being more correlated to the evaluation result.

It is worth mentioning that the preciseness of the multivariate single Gaussian model and univariate GMM were fairly close to the best result (less than 0.1 units). Therefore, we can confirm that the multivariate based concept and the use of GMM are indeed valid and being effective for the fusion of multi-dimensional features, which we believe can be useful for extracting additional hidden knowledge from high dimensional space and cooperate better with wide class varieties.

4 Discussion

4.1 Intermedia function between confidence margin and decision score

In section 2.1, we have used the decision score in (3) as a simplified evaluation of the strength in decision making. However, the relation between the decision score and the final decision strength could be more flexibly treated due to the external bias involved during the decision making in the real practice. For example, if confusion exists in deciding over a scan image, the doctor may favour PUV over MC because PUV is a safer option, which has a second chance to be retested. Therefore, it would be helpful to introduce an intermedia function $\mathcal{T}$ that further refines the decision score S_D and yields a more realistic final decision towards the intended bias.

The regression of function $\mathcal{T}$ could be modelled in any format, which can be learned from a validation dataset after the probability model of the S_D has been trained. Take a linear regression model as an example; the transportation function $\mathcal{T}$ can be presented as:

$$\mathcal{T}(S_D) = m \cdot S_D + c : [-1,1] \mapsto [-1,1] \tag{10}$$

where *m* and *c* are the two constants that represent the transformation rate (a multiplier that indicates the trust of the calculated decision score) and the external bias (which one of the class we prefer more) respectively.

Unfortunately, this concept could not be validated due to the shortage of data samples. However, it is a valid idea and could enhance the preciseness of the confidence calculation.

4.2 The use of adaptive GMM under individual sub-classes

Arguably, as a further extension to (8), each class could also be modelled more realistically with a Gaussian mixture since it would be natural to have multiple subclasses in one class. However, the precise number of the sub-classes and their distributions are unknown. As a solution, we could first use Expectation Maximisation algorithm [11] to model the right distribution of each sub-Gaussian model of any given *K* values, and then selecting the best model set from all the possible *K* values by measuring their information criterions. Accordingly, we could derive the appropriate K_{ω_i} for each class

ω_i with their relevant parameter set $\{\theta_{\omega_i}{}_{j=0\ldots K}\}$, where our previous model $P(x)$ and $P(x|\omega_i)$ can then be further extended to:

$$\begin{cases} W_i = \frac{|[\omega_i]|}{|\Omega|} \\ P(\vec{x}) = \sum_{i=1}^{k} W_i \mathcal{N}(\vec{x} \mid \theta_{\omega_i}{}_{j=1\ldots K_{\omega_i}}) \\ P(\vec{x}|\omega_i) = \mathcal{N}(\vec{x} \mid \theta_{\omega_i}{}_{j=1\ldots K_{\omega_i}}) \end{cases} \quad (10)$$

We have in fact tried this way of modelling with the data set we have. However, we found that the *K* for each class is always equal to 1, which by nature result in a model that has no difference to the model in (8). Nevertheless, we believed that it is still worth to further validate this argument with different data sets in the near future.

5 Conclusions

This paper presented a concept of classification confidence, which reflects a systematic treatment of decision strengths based on posterior probabilities. The decision score function not only indicates the class label with +/− sign but also gives an absolute measurement of the level of confidence and belief to the decision made. The basic definition of confidence score based on a single univariate Gaussian distribution was extended to multivariate Gaussian mixture models. We argued the validity of the confidence score and tested its use in a real-life early pregnancy data set. The test result shows a strong correlation between the MSD feature values and the confidence-based predictions of miscarriage cases vs. PUV cases, indicating the potential use of the confidence level, which can be further explored in a multiple features and multiple classifier fusion framework.

Our future work includes further testing the use of confidence score in other real-life data sets of various kinds, fine tuning the concept as suggested in the Discussion part, and developing confidence-based fusion framework inside a decision support system.

Acknowledgment

The authors wish to thank the Early Pregnancy Department, Queen Charlotte and Chelsea Hospital, Imperial College London in general and Prof Tom Bourne and Dr Jessica Farren in particular for providing the data set for this study.

References

1. C. E. Metz, "Basic Principles of ROC Analysis," in *Seminars in Nuclear Medicine*, 1978.
2. M. Majnik and Z. Bosni, "ROC Analysis of Classifiers in Machine Learning: A Survey," 2011.

3. L. Xue, X. Yang, W. Wu, M. Chen and X. He, "An algorithm based on KNN and improved SVM for licence plate recognition," *Journal of Sichuan University*, vol. 43, no. 5, pp. 1031-1036, 2006.
4. H. Du, Data Mining Techniques and Applications, An Introduction, Cengage Learning, 2010.
5. R. Li, S. Ye and Z. Shi, "SVM-KNN classifier-a new method of improving the accuracy of SVM classifier," *Acta Electronica Sinica*, vol. 30, no. 5, pp. 745-748, 2002.
6. S. Khazendar, H. Al-Assam and H. Du, "Automated Classification of Static Ultrasound Images of Ovarian Tumours Based on Decision Level Fusion," in *Computer Science and Electronic Engineering Conference (CEEC)*, 2014.
7. G. Shafer and V. Vovk, "A Tutorial on Conformal Prediction," *Journal of Machine Learning Research* , pp. 371-421, 2008.
8. J. Wang, P. Neskovic and L. N. , "A Statistical Confidence-Based Adaptive Nearest Neighbor Algorithm for Pattern Classification," in *International Conference on Machine Learning and Cybernetics*, 2006.
9. H. Papadopoulos, V. Vovk and A. Gammermam , "Conformal Prediction with Neural Networks," in *19th IEEE International Conference on Tools with Artificial Intelligence*, 2007.
10. T. Bourne and C. Bottomley, "When is a pregnancy nonviable and what criteria should be used to define miscarriage?," *Fertility and sterility*, pp. 1091-1096 , 2012.
11. T. K. Moon , "The expectation-maximization algorithm," *IEEE Signal Processing Magazine*, vol. 13, no. 6, pp. 47 - 60 , 1996.

Reservoir Computing: Evolution *in materio*'s Missing Link

Matthew Dale[1,3], Julian F. Miller[2,3], Susan Stepney[1,3], Martin A. Trefzer[2,3]

[1]Department of Computer Science, University of York, UK
[2]Department of Electronics, University of York, UK
[3]York Centre for Complex Systems Analysis
md596@york.ac.uk

Abstract. Reservoir Computing (RC) is a computational framework that has the potential to transfer onto any input-driven dynamical system, given two properties are present; a fading memory and input separability. A typical reservoir consists of a fixed network of recurrently connected processing units. When excited, the network maps the incoming temporal signal into its high-dimensional state space. The resulting spatial-temporal state space can then be trained to approximate a vast number of functions. To train the reservoir, a simple readout mechanism is adapted, selecting and combining different reservoir states to form the desired output. Our recent work has demonstrated how this framework can be applied to randomly-formed carbon nanotube composites to solve computational tasks. In this paper, we compare our proposed reservoir computing *in materio* technique to the original evolution *in materio* technique using the *Iris* classification task. The results show that despite the non-temporal nature of the task, reservoir computing *in materio* can outperform evolution *in materio* using the same composites.

Keywords: Material Computation, Evolution *in materio*, Reservoir Computing, Unconventional Computing, Machine Learning Classification

1 Introduction

Research in unconventional computing has shown that many diverse dynamical systems can be exploited to perform computational tasks [1]. Exploiting computation at the substrate-level offers potential advantages over classical computing architectures, such as exploiting physical and material constraints that could offer solutions"for free", or at least computationally cheaper [17].

Computational composites consisting of randomly dispersed carbon nanotubes have shown interesting electrical properties that can be trained to perform a variety of computational tasks [3]. To extract computation from these composites a technique known as evolution *in materio* is applied [14]. Evolution *in materio* attempts to intrinsically evolve physical solutions to computational problems, for example, by manipulating the electrical response of a material to input stimuli. This is achieved by evolving a number of physical parameters such as the value, type and placement of input signals on the material.

Evolution *in materio* is a growing field with new hardware developments, experimental materials and directions appearing from different research groups, e.g. [2, 13]. However, what is classed here as the "*vanilla*" technique has its limitations. For example, the technique typically used in the literature produces a discretised output, represented by voltage samples averaged in an output buffer. In that technique, "programming" is typically done using digital voltage signals, such as square waves, between a limited voltage range (due to hardware limitations). Using such a model one cannot fully exploit the temporal properties of the material, which is argued here to be a rich source of untapped information.

In recent work [6, 7], we have shown that by applying the Reservoir Computing framework to the same materials, we can exploit interesting temporal properties that were previously unused. The new addition of the reservoir framework has enabled temporal problems requiring both dynamic behaviours and memory to be tackled. Here, we investigate how well the new reservoir/material framework compares in performance to the *vanilla* evolution *in materio* technique on a non-temporal task. The task is to perform classification on the Fisher Iris data set across a range of carbon nanotube composites, including a control (a resistor array) and the same material used in [15]. The last section of this paper presents two simple analysis techniques that could provide further understanding as to what reservoir/material properties are being exploited by evolution.

2 Reservoir Computing

Reservoir Computing first emerged as efficient mechanism for training recurrent neural networks and later evolved into a general theoretical model for many dynamical systems [16]. By applying only a relatively simple training mechanism many physical systems have become exploitable unconventional computers (see summary [5]).

A reservoir functions much like a temporal *kernel* [10] and can represent any excitable non-linear medium, given the medium can: (i) create a high-dimensional projection of the input into observable *reservoir states*; and (ii) possess a fading memory of previous inputs and internal states. These prerequisite properties are described by the *separation*, *approximation* [11] and *echo state* properties [8]. With these properties a reservoir can realise any non-linear filter with bounded memory, and with the aid of a trained readout approximate any function.

To interpret and train the material as a reservoir we define the observed reservoir states $x(n)$ of the material as a combination of the *continuous* material and the *discrete* observation function:

$$x(n) = \Omega(\mathcal{E}(W_{in}u(t)) \tag{1}$$

where $\Omega(n)$ is the observation of the macroscopic material behaviour and $\mathcal{E}(t)$ the microscopic material function when driven by the input $u(t)$, multiplied by the input weight matrix W_{in}.

The training of the reservoir readout is typically done using Ridge Regression [9], manipulating the weights W_{out} to reduce the error (Normailised Root Mean Squared Error *NRMSE*) between the training signal $y_{target}(n)$ and the reservoir output $y(n)$:

$$W_{out} = Y_{target} X^T (XX^T + \beta I)^{-1} \tag{2}$$

where I is the identity matrix and β the Tikhonov regularisation parameter. The trained reservoir system is given in (3).

$$y(n) = W_{out} x(n) \tag{3}$$

3 Experimental Platform

The materials used in this experiment consist of various concentrations (w.r.t weight) of single-walled carbon nanotubes, randomly dispersed in an insulating polymer: poly-butyl-methacrylate (PBMA) or poly-methyl-methacrylate (PMMA). Each material is deposited onto a gold/chromium microelectrode array of 12 electrodes to form input-outputs to the material. The electronic properties of the dispersed nanotubes are approximately $\frac{1}{3}$ metallic nanotubes and $\frac{2}{3}$ semiconducting nanotubes. The relative size of the nanotubes (100nm to 1000nm length and diameter between 0.8nm and 1.2nm) is significantly less than the gap between the electrodes (between 100 to 150 µm) suggesting the nanotubes form conductive pathways between the electrodes. The polymer mixed with the nanotubes acts both as a dielectric and as a suspension for the nanotube network. In previous experiments, it has been observed that both a conductive network is formed and a high computational performance is reached around a percolation threshold of 1% carbon nanotubes [6, 7, 12].

To interface the computer with the material, two input/output data acquisition cards are used. Each card is controlled by a MATLAB interface and is set to either input, or read, analogue voltages from the micorelectrode array. To allow each card to communicate with any electrode, a cross-point switch is added allowing evolution to reconfigure and route different input-output combinations.

4 Machine Learning Classification

The Iris data set[1](also known as *Fisher's* Iris data set) is a well-known multivariate data type classification problem that was previously used to benchmark the evolution *in materio* technique [4, 15]. The task is to classify three species of the Iris plant given the four attributes petal and sepal length and width. The Iris data set contains 150 instances, with 50 instances of each class/species. The data set was evenly divided into training and testing sets of 75 randomised instances,

[1]The Iris data set can be found on the UCI Machine learning repository at: https://archive.ics.uci.edu/ml/

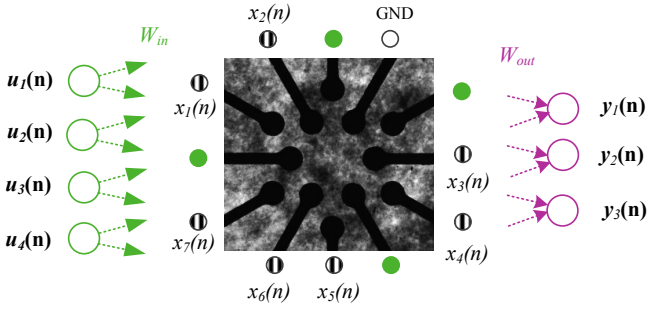


Fig. 1: Input-Output mapping of the task inputs $u(n)$ and observed reservoir states $x(n)$. For this task 4 inputs are required ($u_{1:4}(n)$), each input is multiplied by the input weight matrix W_{in} and applied to an electrode (green). To form the 3 output classes $y_{1:3}(n)$, the reservoir states ($x_{1:7}(n)$) are multiplied by the output matrix W_{out} (the trained readout layer).

containing 25 instances of each class. Each class in the reservoir framework is represented as a separate reservoir output (Fig.1) with a binary value, and each attribute is represented by a floating-point input voltage.

5 Training Reservoir Computers

Each material is deposited onto a microelectrode array, providing electrical inputs and outputs to the system. Using computer-based evolution, the role of each electrode is decided as well as some other associated parameters. For example, evolution decides which electrodes are inputs $u(n)$, or outputs $x(n)$, and evolves a *weight* value for each input (see Fig.1). In addition to the weights, another parameter is evolved called the *leak rate*. The leak rate parameter α is used to match the dynamics of the reservoir to the task input and/or target output. The leak rate parameter is applied post-collection of states, see eq.(4). This turns eq.(3) into eq.(5) – this parameter, however, may have little effect on reservoir performance for non-temporal tasks.

$$\tilde{x}(n) = (1-\alpha)x(n-1) + \alpha\Omega(\mathcal{E}(W_{in}u(t)) \tag{4}$$

$$y(n) = W_{out}\tilde{x}(n) \tag{5}$$

In this experiment we apply an evolutionary strategy of (1+4) of 150 generations, across 20 runs. The implementation of this strategy when combined with Gaussian mutation operators functions similar to a hill-climber algorithm. This is set to reduce the retention effect of degenerative fitness jumps experienced by rapidly changing the configuration parameters.

Training and testing is split into two phases. Using the *training set*, evolution selects appropriate input-output mappings and ridge regression trains the

readout layer. Then in testing, a final evaluation of the material configuration (and trained readout) is carried out using the *test set*.

5.1 Fitness Evaluation

To train the reservoir and evolve a solution, the *NRMSE* between the trained output and the target output is used to define fitness. However, as the task is to classify binary classes – not a time-series output – a threshold mechanism is used to translate the trained outputs into binary classes. To evaluate the accuracy of the classifier, and to conduct a fair comparison to the evolution *in materio* technique, the fitness calculation in [15] is applied. This is provided in eq.(6), evaluating the number of true positives *TP*, true negatives *TF*, false positives *FP* and false negatives *FN* that occur.

$$\text{Fitness} = \frac{TP.TN}{(TP + FP)(TN + FN)} \tag{6}$$

To select a threshold, a simple optimisation loop was used (post-state collection). The best threshold was then attached and stored to the given configuration and reimplemented at the testing phase.

6 Experimental Results

Fig.2 and Fig.3 show the results of both simulated and physical reservoirs on the Iris task. Fig.2 shows that performance varies with respect to carbon nanotube concentration, with concentrations of 0.71% and above outperforming the others, and well above the control material (resistor array). In Fig.2, two 0.71% PMMA materials are shown, one of which is the material used in [15]; which one is unknown. Both outperform the reported *average* test accuracy of 77.1%, with both average *training* and *test* accuracies above 90%.

Fig.3 compares the performance of a physical reservoir (the 1% PBMA material) to three evolved simulated (*in silico*) reservoirs. The similarity in accuracies suggests the material is forming an optimised reservoir and is possibly exploiting the benefits of both a smaller readout and a larger network; much like the sampled version which is only using 7 states out of a possible 50. The performances of these reservoirs also compare favourably to the Cartesian Genetic Programming method stated in [15] (*train*: 97.7%, *test*: 93.6%), with training accuracies typically above 95% and test accuracies averaging between 90-95%. These results therefore suggest our *in materio* method can compete with two optimised *in silico* methods on this task.

The results of this experiment are surprising, showing that applying the reservoir computing framework on a non-temporal task, performance can be increased over *vanilla* evolution *in materio*. The results appear counter intuitive on the surface, as one would assume the direct programming nature of evolution *in materio* would outperform our technique. However, this is not the case. Possible explanations for this increase in performance might be that: (i) stimulating the

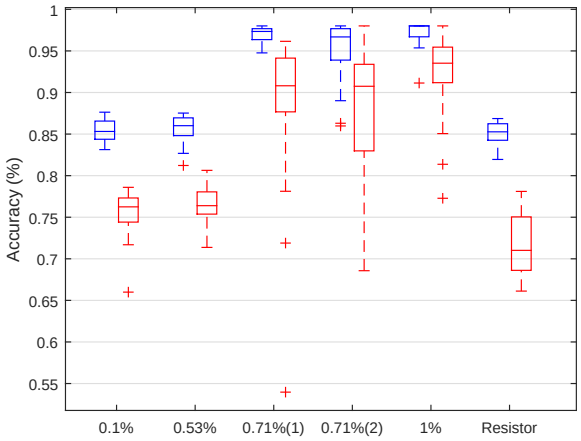


Fig. 2: Experimental results of each material using eq.(6). Each material shows the training accuracy (blue) and test accuracy (red) across 20 evolutionary runs. Two 0.71% PMMA materials are given, with one being the same used in [15]

material with variations of the input causes interesting interactions not present in the *vanilla* technique; (ii) a conductive network might not be present across all electrodes and instead several networks may exist across the array, therefore additional inputs-outputs grant access to each of these networks; (iii) combining the outputs and weighting their importance allows training to exploit the whole material rather than exploiting a single area around a particular electrode (which relates closely to (ii)). These possibilities are suggested here as starting points for further exploration.

7 Analysing Reservoir and Material Characteristics

As part of the investigation three evolved simulated reservoirs (*Echo State Networks* [8]) of different sizes were added for comparison (Fig.3). Each reservoir has three key parameters that were selected through evolution; the *spectral radius*, *input scaling* and *leak rate*. Each parameter is adjusted to tune the dynamical behaviour and memory of the reservoir. In Echo State Networks, the spectral radius determines how fast the influence of the input degrades and the stability of the reservoirs activations. The Input scaling parameter is used to tune the non-linearity of the reservoir and tune the proportional effect the current input has compared to previous activations. The leak rate parameter is used to match the speed of the reservoirs dynamics to the task input and/or output; essentially applying additional filtering to the activation of each node [9].

Fig.4 shows the relationship between parameters and task performance for each evolved network. By visualising the evolved parameters and their relative performance, we can determine the desired dynamics for a given task and suggest what dynamics need to be exploited in the material. From the plots we can see

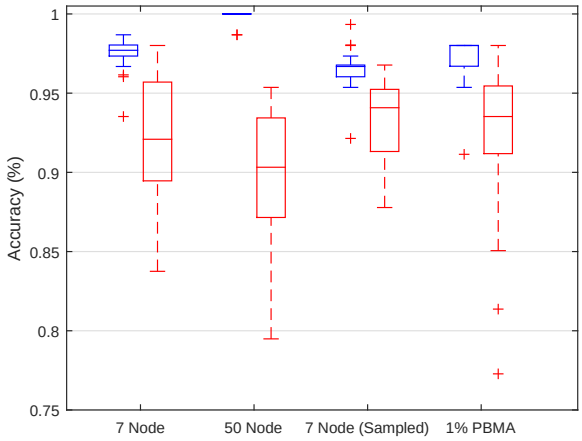


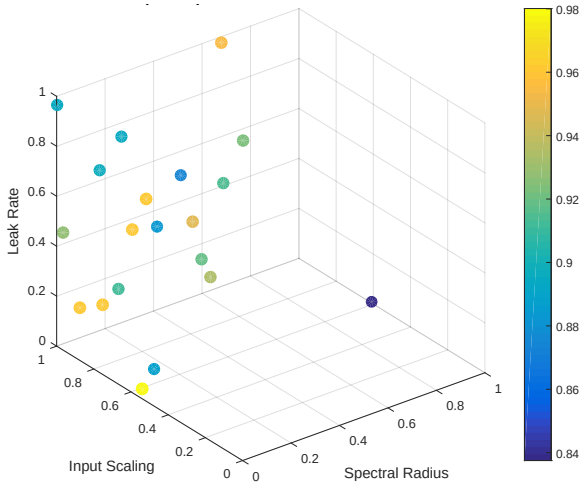
Fig. 3: Experimental results of three simulated reservoirs and the 1% PBMA material using eq.(6). Training accuracy (blue) and test accuracy (red) are given across 20 evolutionary runs.

that each network (independent of size) typically possesses the following for the Iris task:

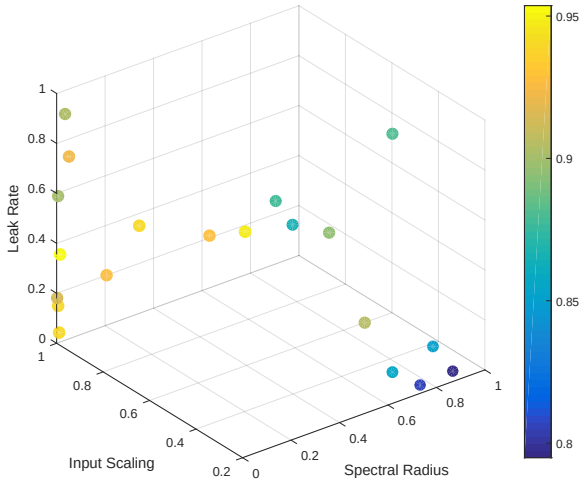
- Low spectral radius; i.e. does not require longer memory – output depends upon current input – and the reservoir is stable.
- Large variation in leak rates; suggesting the parameter does not significantly effect performance – as expected for a non-temporal task.
- Input scaling ≤ 1; neurons are not saturated and sway towards linear dynamics.

These evolved dynamics appear to match the known electrical properties of our materials (both stable and slightly non-linear I-V characteristics). This therefore suggests why the reservoirs perform similarly both *in materio* and *in silico*.

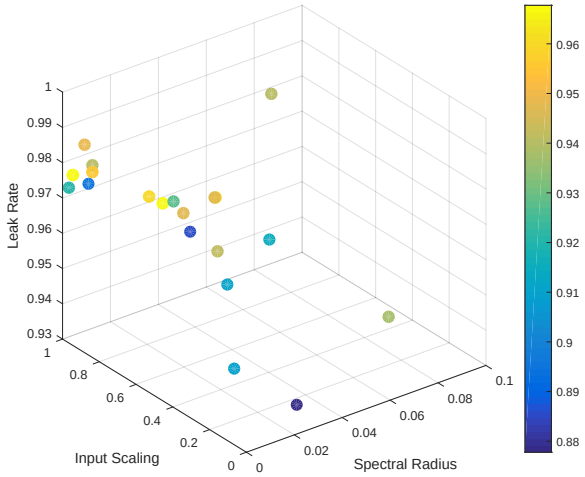
Mapping the dynamical and macroscopic relationships between different architectural reservoir systems can be enlightening, not only to benchmark the systems but to help identify what electrical properties of the material respond well to reservoir-style training. However, identifying exactly what properties in the material are being exploited by evolution is challenging. As discussed in [5], analysing and modelling the nanotube structures and electrical pathways being used is difficult. In Fig.5 we introduce a simple visualisation that highlights areas of interesting activity typically exploited by evolution. The figure displays the electrode arrangement for each material, showing what frequency an electrode is selected as an input (circle size) and what average voltage value is supplied (circle colour) across different evolutionary runs.



(a) 7 node network

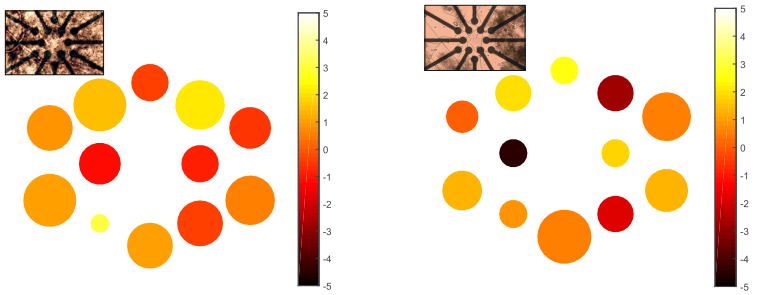


(b) 50 node network



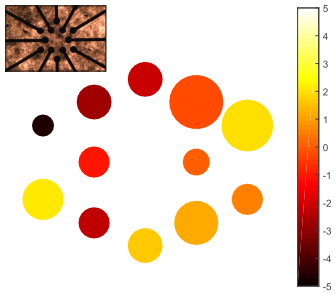
(c) 50 node (sampled) network

Fig. 4: Each plot displays 20 evolved Echo State Networks with network sizes of; (a) 7 nodes, (b) 50 nodes, and (c) 50 nodes with only 7 nodes being used to train/form the output. The parameters under evolution are input scaling (x axis), leak rate (y axis) and spectral radius (z axis). The colour map shows test accuracy of each evolved parameter set.

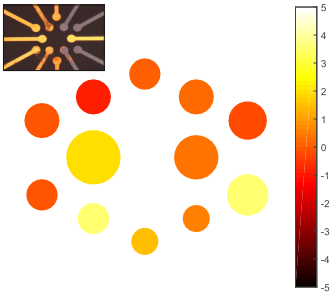


(a) 0.53% PMMA

(b) 0.71% PMMA(1)



(c) 0.71% PMMA(2)



(d) 1% PBMA

Fig. 5: Evolved circular electrode plot: Each plot shows how frequent an electrode is selected as an input (circle size) for each material, and the average input weight (circle colour), i.e. average voltage intensity, converged upon through evolution.

8 Conclusion

The results of this investigation are intriguing as they show that adding the reservoir framework – which enables the full exploitation of temporal information and some level of general-purpose functionality – there is no degradation in performance relative to the *vanilla* technique. We demonstrated this across several materials, including a control (a resistor array) and the same material used in the comparison work. This initial experiment shows that reservoir computing *in materio* can outperform the evolution *in materio* technique and match the performance of *in silico* techniques.

The final section of this work presented visualisation tools to aid understanding of the required task dynamics. These tools can be used to identify suitable tasks for the material, and identify regions of interesting activity/conductivity in the material often exploited by evolution.

Acknowledgements. Matthew Dale is funded by a Defence Science and Technology Laboratory (DSTL) PhD studentship. The authors thank the EU NASCENCE Project for providing the SWCNT materials used in this work.

References

1. A. Adamatzky, L. Bull, and B. D. L. Costello. *Unconventional computing 2007*. Luniver Press, 2007.
2. S. Bose, C. Lawrence, Z. Liu, K. Makarenko, R. van Damme, H. Broersma, and W. van der Wiel. Evolution of a designless nanoparticle network into reconfigurable boolean logic. *Nature nanotechnology*, doi:10.1038/nnano.2015.207, 2015.
3. H. Broersma, J. F. Miller, and S. Nichele. Computational matter: Evolving computational functions in nanoscale materials. In A. Adamatzky, editor, *Advances in Unconventional Computing*, volume 2, pages 397–428. Springer, 2017.
4. K. D. Clegg, J. F. Miller, M. K. Massey, and M. C. Petty. Practical issues for configuring carbon nanotube composite materials for computation. In *ICES 2014, IEEE International Conference on Evolvable Systems*, pages 61–68. IEEE, 2014.
5. M. Dale, J. F. Miller, and S. Stepney. Reservoir computing as a model for in-materio computing. In A. Adamatzky, editor, *Advances in Unconventional Computing*, volume 1, pages 533–571. Springer, 2017.
6. M. Dale, J. F. Miller, S. Stepney, and M. A. Trefzer. Evolving carbon nanotube reservoir computers. In *Unconventional Computation and Natural Computation: 15th International Conference, UCNC 2016, Manchester, UK, July 11-15, 2016*, pages 49–61. Springer International Publishing, 2016.
7. M. Dale, J. F. Miller, S. Stepney, and M. A. Trefzer. Reservoir computing in materio: An evaluation of configuration through evolution. In *IEEE International Conference on Evolvable Systems (ICES), (submitted)*. IEEE, 2016.
8. H. Jaeger. The "echo state" approach to analysing and training recurrent neural networks-with an erratum note. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148:34, 2001.
9. M. Lukoševičius. A practical guide to applying echo state networks. In *Neural Networks: Tricks of the Trade*, pages 659–686. Springer, 2012.
10. M. Lukoševičius, H. Jaeger, and B. Schrauwen. Reservoir computing trends. *KI-Künstliche Intelligenz*, 26(4):365–371, 2012.
11. W. Maass. Liquid state machines: motivation, theory, and applications. *Computability in context: computation and logic in the real world*, pages 275–296, 2010.
12. M. Massey, A. Kotsialos, F. Qaiser, D. Zeze, C. Pearson, D. Volpati, L. Bowen, and M. Petty. Computing with carbon nanotubes: Optimization of threshold logic gates using disordered nanotube/polymer composites. *Journal of Applied Physics*, 117(13):134903, 2015.
13. M. Massey, A. Kotsialos, D. Volpati, E. Vissol-Gaudin, C. Pearson, L. Bowen, B. Obara, D. Zeze, C. Groves, and M. Petty. Evolution of electronic circuits using carbon nanotube composites. *Scientific Reports*, 6, 2016.
14. J. F. Miller and K. Downing. Evolution in materio: Looking beyond the silicon box. In *NASA/DoD Conference on Evolvable Hardware 2002*, pages 167–176. IEEE, 2002.
15. M. Mohid, J. F. Miller, S. Harding, G. Tufte, O. R. Lykkebø, M. K. Massey, and M. C. Petty. Evolution-in-materio: Solving machine learning classification problems using materials. In *PPSN XIII, Parallel Problem Solving from Nature*, pages 721–730. Springer, 2014.
16. B. Schrauwen, D. Verstraeten, and J. Van Campenhout. An overview of reservoir computing: theory, applications and implementations. In *Proceedings of the 15th European symposium on artificial neural networks*. Citeseer, 2007.
17. S. Stepney. The neglected pillar of material computation. *Physica D: Nonlinear Phenomena*, 237(9):1157–1164, 2008.

Part III

Extended Abstracts

A Convolution Kernel for Sentiment Analysis using Word-Embeddings

James Thorne

Department of Computer Science
University of Sheffield
jthorne1@sheffield.ac.uk

Abstract. Accurate analysis of a sentence's sentiment requires understanding of how words interact to convey emotion. Current works use the sentence's parse tree and recursively compute sentiment of the constituent phrases. This approach is expensive and requires a human to annotate all subtrees of a sentence. We examine how using a lexical similarity kernel can leverage word-embeddings, generated in an unsupervised manner, and capture the interactions between all word units. In our evaluation, we find that this type of kernel performs with comparable accuracy to the state-of-the-art for polar sentiment analysis. We examine weighting features by node-depth in the sentence's parse tree; however, this yielded unsatisfactory results. The sentence's parse tree must be leveraged to attain better sentiment analysis and we are confident that this kernel provides a framework that can be extended in future to better make use of this information.

Keywords: Natural Language Processing, Support Vector Machine, Sentiment Analysis, Kernel, Convolution, Tree Kernel, Embeddings

1 Introduction

Sentiment analysis is the task of identifying and extracting opinions and emotion from natural language. A large body of sentiment analysis work in the consumer domain been motivated by the need to understand the opinions, attitudes and emotions of customers.

A word's sentiment is often disconnected from the semantic meaning of individual word units, requiring examination of a sentence as a whole. This is highlighted with a simple example: the two sentences *"The car's steering is unpredictable"* and *"The movie's plot was unpredictable"* both use *unpredictable* as a valence shifter. However, one sentence is positive and the other is negative.

Recently there has been a large body of work focused on generation of distributional word representations called embeddings (see [1–3]): real-valued vectors which describe the meaning of word units. This both allows the similarity of words to be inferred through standard measures such as the cosine distance and allows the meaning of unseen words to be inferred through the context in which that word appears in. Embeddings abstract away from word surface forms and

have been used as features in classification tasks such as named entity recognition [4] and sentiment analysis [5].

This report investigates the modelling of sentiment analysis as a kernel-based classification task. Kernel methods are a powerful machine learning technique that allow features to be modelled in a high dimensional, implicit, metric space. We incorporate both word embeddings learned through word2vec [2] and features from the sentences parse tree in a convolution kernel that is an extension of the lexical semantic kernel [6]. The working assumption of this model is that words which appear deeper in a sentence's parse tree have less weight in the semantic orientation of the sentence.

The differentiating factor in this work is that the word embeddings are learned in an unsupervised manner from a neutral 'out-of-domain' data set. This is contrasted to the leading state-of-the-art method by Socher et al[7] which learns word embeddings in a semi-supervised environment and makes use of human-annotated sentiment labels in training the embeddings.

We find that the addition of node depth as weights in the Lexical Semantic Tree Kernel (LSTK) adversely affects the model's accuracy. This reduces the accuracy below that of other sentiment classification methods such as Recursive Neural Tensor Networks [7] and Paragraph Vector [8]. However, we find that by simply incorporating embeddings into the Lexical Semantic Kernel, we are able to achieve an accuracy score that is comparable to the vector-average of embeddings in [7] despite only using embeddings generated on an entirely out-of-domain dataset.

2 Method

2.1 Word Embeddings and Word2Vec

Word embeddings are low-rank factorisations of a word-context Pointwise Mutual Information (PMI) matrix. The inner dimensions of the factor matrices correspond to word senses observed in training. The state of the art method, Word2vec [2, 9] computes these factorised matrix components implicitly [10].

2.2 Lexical Semantic Tree Kernel

This project introduces the Lexical Semantic Tree Kernel (LSTK). The LSTK incorporates the depth of words in a sentence's constituency tree as feature weights for a Lexical Semantic Kernel (LSK) [6].

A general LSK is defined as a pairwise similarity convolution kernel over a pair of documents. To incorporate real-valued embedding vectors we alter the kernel to use the cosine similarity of embeddings instead of graph distance in an ontology.

$$K(x, y) = \sum_{w_1 \in x} \sum_{w_2 \in y} \lambda_{w_1} \lambda_{w_2} \sigma(w_1, w_2) \tag{1}$$

We define the LSTK over a pair of sentence parse trees (T_1,T_2) as the word similarity score between the words at leaf nodes (words) weighted by the node depth. This weighting was chosen under the assumption and working hypothesis that valence shifters which have a lower depth in the tree (i.e. they are closest to the root node) have a higher influence on a sentence's sentiment orientation over the nodes with a higher depth.

$$K(T_1, T_2) = \sum_{w_1 \in \text{leaf}(T_1)} \sum_{w_2 \in \text{leaf}(T_2)} \frac{1}{\text{depth}(w_1) \cdot \text{depth}(w_2)} \sigma(w_1, w_2) \qquad (2)$$

Optimisation: The distributive property of the dot-product operation is exploited to reduce the complexity of the LSK to linear with respect to the sentence lengths of x and y:

$$K(x, y) = \sum_{w_1 \in x} \sum_{w_2 \in y} \lambda_{w_1} \lambda_{w_2} \sigma(w_1, w_2) = \sigma(\sum_{w_1 \in x} \lambda_{w_1} w_1, \sum_{w_2 \in y} \lambda_{w_2} w_2) \qquad (3)$$

3 Results

LSK and LSTK kernels were tested using the Stanford Sentiment Treebank [7] dataset. These kernels were applied to the sentence-level polar and fine-grained sentiment analysis of movie reviews. For the fine-grained task, 8544 training and 2210 test samples were annotated with a continuous score which captured the strength and polarity of the sentiment; this score is quantised into five classes (`very negative`, `negative`, `neutral`, `positive` and `very positive`). We complete the polar task in the same manner as [7] to enable comparison: the `very positive` and `positive` classes were merged (likewise for the negative classes) and the `neutral` class was dropped. This reduced the number of training samples to 6920 and the number of test samples to 1821.

The evaluation was conducted using the F1-score classification metric: the harmonic mean of the classifier's precision (ratio of false positives) and recall (ratio of false negatives). In the fine-grained dataset, class numbers were not balanced (ranging between 271 and 604 samples per class). The summary statistics presented are weighted average scores based on the number of test samples.

Adding the depth of nodes as weights in the LSTK did not improve the accuracy over our baseline. This allows us to reject our working hypothesis and suggests that depth of nodes in a sentence's constituency tree alone does not play an important factor in determining a word's weight for comparing similarity of two sentences.

Neither of the new techniques was able to improve performance for the fine-grained sentiment analysis task. While the current state of the art RNTN performs with a mere 45% accuracy [7], it better captures the interplay between phrases within a sentence and presents significantly higher accuracy than the techniques presented in this work.

Method	Fine-Grained			Polar		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Bag of Words SVM [7]	-	-	0.407	-	-	0.794
RNTN [7]	-	-	0.457	-	-	0.854
VecAvg [7]	-	-	0.327	-	-	0.801
LSK	0.4989	0.4226	0.3394	0.8025	0.8018	0.8017
LSTK	0.4797	0.3898	0.3003	0.7799	0.7796	0.7795

Table 1. Sentence-level Sentiment analysis task over the Stanford Sentiment Treebank

One of the key differences between how the RNTN and VecAvg methods were trained in [7] is that the embeddings are generated in a supervised environment that exploits the human-annotated sentiment for all sub-phrases within a sentence. We contrast this against the LSK and LSTK kernel-based classifiers, which are trained only at the root level of the sentence's parse tree using embeddings that only capture semantics. While this is sufficient to generate equivalent scores for the polar task, this is not sufficient for the fine-grained analysis.

4 Conclusions

This report evaluated the performance of a kernel for sentiment analysis which weighted lexical similarity with the depth of the words in the parse tree. The working hypothesis was that words which appear deeper within a sentence's constituency tree have less weight for shifting the valence of sentences. The inclusion of this weight reduced the performance for polar sentiment analysis, leading us to reject this initial hypothesis.

The LSTK and RNTN approaches are suitable when parse trees of the sentence are available. The generation of a parse tree adds extra expense to the sentiment analysis and will introduce an additional error term. While the baseline LSK presents an advantage of not requiring a parse tree, it appears that this information is necessary to capture the finesse of the statement and improve F1 scores beyond the 'ceiling' of 0.8 for the polar task.

Even though word2vec does not incorporate sentiment into the generation of the word embeddings, using these out-of-domain unsupervised embeddings on the LSK provided satisfactory results for the polar classification. This is comparable with the baseline VecAvg without the need to train sentiment specific word embeddings.

Future work will in investigate whether sentiment-specific-word-embeddings [5] provide an increase of performance for sentiment analysis given a sentence's parse tree. Additionally, there would be merit in conducting an investigation of whether a more complex tree kernel that captures composition between different nodes rather than only evaluating leaf nodes.

References

[1] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*, ACM, 2008, pp. 160–167.

[2] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[3] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: `http://www.aclweb.org/anthology/D14-1162`.

[4] J. Turian, L. Ratinov, and Y. Bengio, "Word representations: A simple and general method for semi-supervised learning," in *Proceedings of the 48th annual meeting of the association for computational linguistics*, Association for Computational Linguistics, 2010, pp. 384–394.

[5] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin, "Learning sentiment-specific word embedding for twitter sentiment classification.," in *ACL (1)*, 2014, pp. 1555–1565.

[6] R. Basili, M. Cammisa, and A. Moschitti, "Effective use of wordnet semantics via kernel-based learning," in *Proceedings of the Ninth Conference on Computational Natural Language Learning*, Association for Computational Linguistics, 2005, pp. 1–8.

[7] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank," in *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, Citeseer, vol. 1631, 2013, p. 1642.

[8] Q. V. Le and T. Mikolov, "Distributed representations of sentences and documents," *ArXiv preprint arXiv:1405.4053*, 2014.

[9] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *ArXiv preprint arXiv:1301.3781*, 2013.

[10] O. Levy and Y. Goldberg, "Neural word embedding as implicit matrix factorization," in *Advances in Neural Information Processing Systems*, 2014, pp. 2177–2185.

Acknowledgements

This report summarises work from part of an undergraduate project submitted to the Natural Language Processing module at the University of York. Thank you to Suresh Manandhar for facilitating this course and assisting with the runtime optimisation.

An Analysis Approach for Context-Aware Energy Feedback Systems

Fatima Abdallah, Shadi Basurra, and Ali Abdallah

School of Computing and Digital Technology, Birmingham City University, Birmingham B4 7XG, UK
fatima.abdallah@bcu.ac.uk, shadi.basurra@bcu.ac.uk, ali.abdallah@bcu.ac.uk

Abstract. Several energy systems have been developed and studied to help occupants reduce energy usage by providing feedback about their consumption. But recently, a major challenge has emerged about how to enable users to make informed energy efficiency decisions based on consumption feedback. This is because existing systems only present abstract consumption data that are not related to the surrounding energy consumption context. This paper proposes a novel energy data analysis approach which leverages context-awareness to support users to take actions that improve energy efficiency. The approach consists of two stages: multidimensional analysis followed by case-based reasoning. The anticipated output of the analysis approach will be understandable and actionable feedback that helps occupants control their energy consumption.

Keywords: Context-Aware Applications, Energy Consumption Feedback Systems, Multidimensional Analysis, Case-based Reasoning

1 Introduction

The world concern has been more and more concentrated on using energy efficiently since greenhouse gas emissions are expected to increase between 25-90% by 2030 [3]. The European Commission considers that end users will have to play a major role in reducing energy consumption [7]. For this purpose, several energy systems exist to help users reduce their usage by providing information about their energy consumption [2]. However, the information that is provided by existing systems is not enough to inform energy efficiency decision making. This is because these systems only display abstract consumption data which do not allow the users to interpret and understand their consumption [10]. Therefore, an effective energy system needs to integrate energy data analysis capabilities which include: detecting energy waste, providing the reason of the waste, and helping to take appropriate actions [13].

In order to overcome this limitation, we propose the use of context awareness in energy feedback systems. Context awareness involves 3 stages that are referred to as the context life cycle [12]: 1) *context acquisition* which involves collecting different contextual data from physical and virtual sensors, 2) *context*

modelling that is putting the collected data in appropriate repositories, and 3) *context reasoning* which involves analysing the collected context data to provide customized services to the users. Context awareness helps provide understandable and actionable feedback instead of giving abstract feedback. The expected feedback may be of two forms: 1) detailed reports that consist of rich feedback that is supported by context data, and 2) description of energy waste incidents and recommendations that help overcome the waste.

In order to get the desired feedback, we propose an analysis approach that models and interprets energy consumption data using energy related context data. The approach combines multidimensional analysis which integrates the different context data with the consumption data, and case-based reasoning that is used to detect energy waste scenarios and suggest the appropriate energy efficiency advice.

This document is organized as follows: Section 2 highlights limitations of existing energy systems, Section 3 explains the proposed analysis approach with examples of understandable and actionable feedback, and finally Section 4 presents the future work.

2 Energy Systems Related Work

To make occupants aware of their energy consumption, several Energy Consumption Feedback (ECF) systems have been proposed and studied (such as [9]). But recently, a major challenge has emerged about how to enable users to make informed decisions based on consumption feedback [4]. In his investigation into the best way to present electricity feedback, Karjalainen [10] found that although people are motivated to conserve electricity, they are short of information that is needed to take the most appropriate action. Similarly, users of a pilot feedback system reported that they need more context to understand energy usage or take actions [6]. These studies indicate that merely showing the consumer the amount of energy they are using may not be enough to make them understand their consumption and take action to control it.

Another type of energy systems, besides ECF systems, are Energy Management Systems (BEMS) which were thoroughly reviewed in [5]. The review identifies the main components of an ideal BEMS which is expected to monitor energy consumption, sense environmental conditions, detect user presence and preferences, and ultimately help reduce energy consumption. These specifications show that context awareness is one of the core parts of energy systems.

Therefore, there is a need for a context-aware analysis approach that enhances the feedback compared to typical energy feedback systems, and leverages a variety of context data that affects energy consumption.

3 The Energy Analysis Approach

As a first step for context-aware applications development and in order to use context effectively we have defined energy consumption context for context-aware

energy feedback systems in a previous work [1]. The work has found that energy consumption context consists of *User Context*, *Appliances Context*, and *Environment Context* (Fig. 1). These context data are, along with the energy consumption data, the input for the proposed analysis approach (Fig. 2). The context and consumption data can be collected by a smart home technology (sensors) and smart meters.

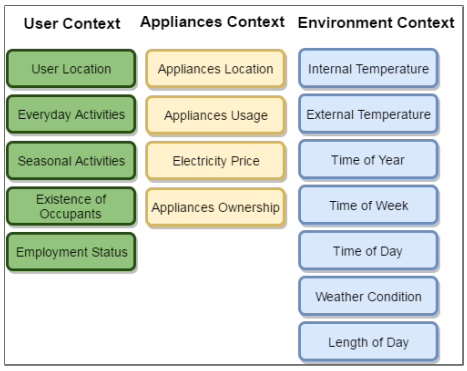


Fig. 1: Elements and categories of the energy consumption context [1]

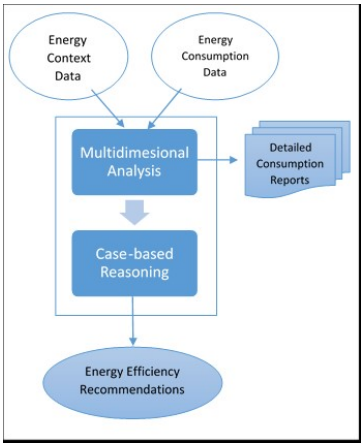


Fig. 2: The proposed energy data analysis approach

Since these data are from different sources, with different formats, and of huge amounts, we propose Multidimensional Analysis (MDA) as the first stage of the analysis approach. MDA can be used to provide detailed consumption reports consisting of rich feedback that is the consumption data supported by the energy consumption context. Therefore, the user can view the consumption data disaggregated in multiple dimensions: room level, user activity level, time dimensions levels (workdays, holidays, weekends), appliances level, or based on outside temperature and weather. These details help the users gain a better understanding of their energy usage, hence, determine practices/conditions that mostly affect it. A similar MDA approach was proposed in [8], which presents a system design that leverages MDA for building management systems to reduce energy consumption. They showed that MDA can support decision-making and provide actionable information for users by combining data from sensors, data from building management systems and data from building information modelling tools [8].

Although detailed reports provide more understanding of consumption, it may not be effective to inform decision-making and implement energy efficiency. Therefore, an automated reasoning method is required to analyse the collected data. For this purpose, the second stage of the analysis approach is a Case-Based

Reasoning (CBR) that detects energy waste, and suggests energy efficiency recommendations. CBR is an artificial intelligence methodology that solves problems based on past experience. Given a current problem with no solution, CBR remembers problems similar to the current situation, and adapts previous solutions to new problems. The adapted solution is then evaluated in order to improve the system's experience [11]. Therefore, through context awareness, CBR detects energy waste incidents by retrieving similar cases based on similar context, and suggests adapted solutions based on the current context. In a similar way, CBR was used for a context-aware healthcare system in [14] and have shown its adaptability compared to traditional rule-based reasoning methods.

The suggested CBR analysis can detect the following energy waste incident: *"The television was left on standby mode in the childrens room between 9pm and 7am while the kids were sleeping. You may save £1 a month if you turn it off before they sleep"*. A similar energy waste case may involve a different device and may be in a different location, at a different time, or with a different occupant activity. CBR is able to detect these similar cases, calculate the amount of wasted energy and forward a notification to the user. Another example that include solution adaptation is as follows: *"The washing machine was turned ON from 8am to 9am yesterday while there was 5 occupants at home and the electricity price was high! You may turn ON the machine between 12pm and 2pm on weekdays when everybody is out of home and the electricity price is low. This will save you £3 a month"*. The suggested solution is to schedule the washing machine in a time when the electricity price is low and the occupants are out of home. Therefore, if a similar case is detected, CBR will search for a suitable time to turn ON the device thus avoiding the detected energy waste.

The difference between a typical CBR system and the proposed analysis approach is in the query initialization. In a typical CBR system, the query is initiated by the user of the system. However, in a context-aware ECF system, which is deployed in a smart home environment, the sensors and smart meters data are streamed continuously. For this purpose, MDA serves as the repository that unifies the formatting of the collected heterogeneous data, synchronizes the data in a specific time interval, and provides the query, which contains all the measurements in the current time, to the CBR for energy waste detection.

4 Future Work

The future work of this proposal is to collect the data needed to test the approach. Real data will be collected through a commercial IoT[1] smart home tool. The data include individual devices energy consumption, occupants motion and presence, temperature, etc. In addition, a smart home simulation software will be used to produce other kinds of scenarios. The simulation tool generates the heating and lighting energy consumption of the house along with sensors data such as temperature and brightness. The approach will then be implemented to test its efficiency and compare it with previous systems.

[1] Internet of Things

References

1. Abdallah, F., Abdallah, A., Basurra, S.: Defining Context for Home Electricity Feedback Systems. In: Proceedings of 2nd International Conference on Zero Carbon Buildings Today and in the Future. Birmingham City University, Birmingham, UK (2016)
2. Aman, S., Simmhan, Y., Viktor K. Prasanna: Energy Management Systems : State of the Art and Emerging Trends. IEEE Communications Magazine 51(1), 114–119 (2013)
3. Barker, T., Bashmakov, I., Bernstein, L., et al.: Technical Summary. In: Metz, B., Davidson, O.R., Bosch, P.R., Dave, R., Meyer, L.A. (eds.) Climate Change 2007: Mitigation. Contribution of Working Group III to the Fourth Assessment Report of the Intergovernmental Panel on Climate Change. Cambridge University Press, Cambridge, UK (2007)
4. Castelli, N., Stevens, G., Jakobi, T., Schönau, N.: Switch off the light in the living room, please!-making eco-feedback meaningful through room context information. In: 28th EnviroInfo Conference. pp. 589–596. Oldenburg, Germany (2014)
5. De Paola, A., Ortolani, M., Lo Re, G., Anastasi, G., Das, S.K.: Intelligent management systems for energy efficiency in buildings: A survey. ACM Computing Surveys 47(1), 13:1–12:38 (2014)
6. Erickson, T., Li, M., Kim, Y., Deshpande, A., Sahu, S., Chao, T., Sukaviriya, P., Naphade, M.: The dubuque electricity portal: Evaluation of a city-scale residential electricity consumption feedback system. In: Proceedings of the 13th SIGCHI Conference on Human Factors in Computing Systems. pp. 1203–1212. ACM, Paris (2013)
7. European Commission: Energy Efficiency Plan 2011. Tech. rep., European Commission (2011), http://ec.europa.eu/clima/policies/strategies/2050/docs/efficiency_plan_en.pdf, accessed: 23 May, 2016
8. Gökçe, H.U., Gökçe, K.U.: Multi dimensional energy monitoring, analysis and optimization system for energy efficient building operations. Sustainable Cities and Society 10, 161–173 (2014)
9. Hargreaves, T., Nye, M., Burgess, J.: Making energy visible: A qualitative field study of how householders interact with feedback from smart energy monitors. Energy policy 38(10), 6111–6119 (2010)
10. Karjalainen, S.: Consumer preferences for feedback on household electricity consumption. Energy and Buildings 43(2-3), 458–467 (2011)
11. López, B.: Case-based reasoning: a concise introduction. In: Synthesis Lectures on Artificial Intelligence and Machine Learning, pp. 1–103. Morgan & Claypool Publishers (2013)
12. Perera, C., Zaslavsky, A., Christen, P., Georgakopoulos, D.: Context aware computing for the internet of things: A survey. IEEE Communications Surveys & Tutorials 16(1), 414–454 (2014)
13. Piette, M.A., Granderson, J., Wetter, M., Kiliccote, S.: Intelligent Building Energy Information and Control Systems for Low-Energy Operations and Optimal Demand Response. IEEE Design & Test of Computers 29(4), 8–16 (2012)
14. Yuan, B., Herbert, J.: Context-aware hybrid reasoning framework for pervasive healthcare. Personal and Ubiquitous Computing 18(4), 865–881 (2014)

Systematic Testing for Distributed Systems

Michael Walker and Colin Runciman

Department of Computer Science, University of York, UK
{msw504,colin.runciman}@york.ac.uk

Abstract. We propose a method of applying systematic concurrency testing (SCT) to distributed systems. SCT enables us to verify assertions for many possible executions of a concurrent program, varying according to scheduling decisions. Distributed systems are similar to concurrent programs in the message-passing style, but messages sent over the network may be lost, re-ordered, or duplicated. Prior approaches avoid these problems by assuming that communications are reliable. Our approach does not have this limitation, and allows a variety of network failure cases to be tested.

Keywords: concurrency, distributed systems, message passing, networking, schedule bounding, swarm testing, systematic concurrency testing.

1 Introduction

Testing concurrency is traditionally difficult due to the scheduler. A program can be run multiple times and produce multiple results, whereas effective testing fundamentally assumes *determinism*. Systematic concurrency testing (SCT) is a family of techniques[1, 2], all with the same general aim: to try to find bugs in concurrent programs, more reliably than just running a program several times.

Distributed systems can be thought of as concurrent programs using message passing as the only communication mechanism, with no shared memory. Erlang is an example of a highly concurrent language using this model[3]. When all communication happens through message passing, in principle it doesn't matter how the threads or processes are assigned to machines.

Protocols such as TCP ensure reliable in-order message delivery in the event of network disruption. Protocols such as UDP do not, and message loss can occur. TCP is the most common protocol used on the Internet, but sometimes UDP is used in cases where the overhead of TCP is great enough to offset the cost of occasional message loss, or when different guarantees are desired. A typical example is real-time online games, where the need for low latency trumps the cost of occasional data loss.

Prior work in this field, by Deligiannis *et al.*[4, 5], assumes reliable in-order message delivery. Their P# work is an extension of C# which allows abstracting away the network layer and testing different interleavings. By modelling network behaviour explicitly, we can do away with this reliability assumption at the cost of some additional complexity.

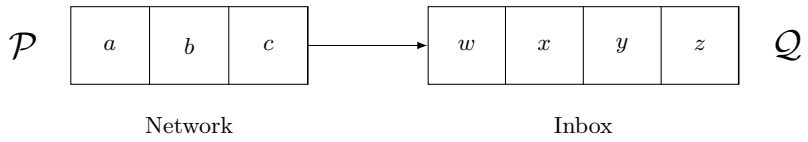


Fig. 1. The directed network link from process $\mathcal{P}$ to process $\mathcal{Q}$.

The paper is organised as follows: Section 2 introduces the network model, showing how it unifies network nondeterminism with scheduler nondeterminism; Section 3 discusses how to guide testing towards diverse network conditions; Section 4 shows a simple domain-specific language we use to express processes; and Section 5 summarises the current state and points to future work.

2 Modelling the Network

SCT techniques assume that any nondeterminism in a program comes solely from the scheduling decisions made. If sending a message from one process to another may nondeterministically fail, that assumption is violated. A similar problem arises in multi-core concurrency when testing *relaxed memory*, as writes to shared memory become visible to other threads nondeterministically. Zhang *et al.*[6] proposed an elegant approach in their 2015 paper, which solves the problem by introducing additional *shadow threads*. Shadow threads are threads which do not exist in the actual program, but are introduced to control other sources of nondeterminism: executing a shadow thread corresponds to that action happening. We can use the same technique here.

In Figure 1, we show the directed network link from process $\mathcal{P}$ to process $\mathcal{Q}$. It is modelled as two queues, where one is the "network" and one is the "inbox". Messages on the network can be interfered with, whereas those in the inbox cannot.

To model the network behaviour, we introduce four shadow threads for each directed network link: **commit** dequeues from the network and enqueues in the inbox; **drop** deletes the head message in the network; **swap** swaps the two head messages in the network; **dup** duplicates the head message in the network.

More complex network behaviours can be built up from these. For example, the network queue can be re-ordered to $\langle c, a \rangle$ by the sequence $\langle$**swap, commit, swap**$\rangle$. By introducing additional threads to model the network behaviour, we once again have a single source of nondeterminism: the scheduler.

3 Guiding the Testing

By introducing four shadow threads for every directed network link (or eight for every pair of processes), we greatly expand the space of possible executions. This makes a *complete* approach to testing infeasible, and poses difficulty for

naive random testing. With no special consideration, the shadow threads—as a whole—are far more likely to be scheduled than the actual program threads. This prolongs execution and causes a great amount of network disruption. We can tackle this problem through two techniques: biassing and bounding[7–9].

- We can *bias* execution away from the shadow threads, by giving every thread a probability of being chosen at a scheduling point, weighted towards program threads.
- We can *bound* the maximum amount of interference (for example, how far any one message can be re-ordered), and forbid any operations which exceed the chosen bounds.
- In addition, some combinations of network disruptions can simply be forbidden, as they do not lead to a new state. For example, ⟨**swap, swap**⟩, ⟨**dup, swap**⟩, or ⟨**dup, drop**⟩.

It is not obvious which probabilities are the *correct* probabilities for biassing, so picking sensible defaults is tricky. Swarm testing[10] tackles a similar problem, the problem of deciding which features to enable in a fuzz tester. In the swarm testing case, it was found that trying different random configurations of enabled features was more effective than trying to hand-tune a single configuration. We have preliminary results indicating a similar approach works well here, with multiple collections of randomly-chosen probabilities performing better than a single hand-tuned set.

The interaction between biassing and schedule bounding can produce interesting executions. For example, a high probability of message loss combined with a low loss bound would tend to manifest as an initial burst of message loss followed by relatively stable execution. This must be investigated, as some combinations may correspond to known network failure cases.

4 Expressing Processes

We have a simple embedded domain-specific language (EDSL), implemented in Haskell, for expressing distributed systems. Our implementation is based on three primitives: **await** blocks until a message has been received or a timeout elapses; **send** sends a message to a named process; **halt** terminates the current process.

Below is a simple example of a process that echoes every received message:

```
echoServer = do
  await >>= \case
    Just (procId, msg) -> send procId msg
    Nothing -> skip
  echoServer
```

The model assumes all processes are defined at system start. Process names can be communicated, but processes cannot be created or started during execution. It is a limited model, but we think it is sufficient to express interesting distributed algorithms and protocols.

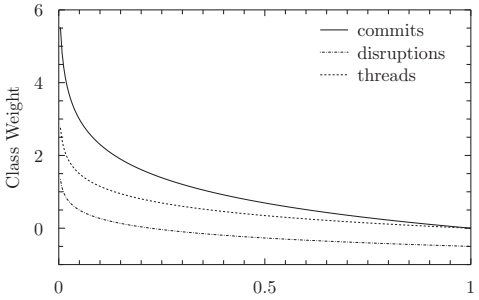


Fig. 2. Class weighting distributions.

5 Current State and Future Work

This work is in its infancy. We have a proof-of-concept implementation showing that the approach is feasible, with few concrete results so far.

Our implementation uses both the swarm testing-like approach to try executions with different probabilities of network disruption, and bounding. We found that assigning probabilities to classes of actions (running a process, dropping a message, etc) and then deriving probabilities for all the actual threads from these to be more effective than assigning a random probability to each thread. We constrain the probabilities so that only the **swap**, **dup**, and **drop** shadow threads may have a probability of zero. Figure 2 shows the weighting distributions for each class of action, where a weight below zero disables the class entirely. Weights are chosen by uniformly picking a value in the range (0,1], and looking up the corresponding weight. Weights are normalised and then transformed into a discrete probability distribution using the condensed table approach[11].

We found that bounding is necessary to ensure termination, as combinations of network actions can easily lead to an unbounded number of messages being produced.

We have applied our technique to Kontiki[12], a Haskell implementation of the Raft consensus algorithm[13], which we seeded with faults. Our technique appears to be effective in finding these faults, but a more careful investigation is called for.

We plan to investigate:

- How fault-tolerant "fault-tolerant" algorithms actually are.
- Do protocols with weaker-than-TCP consistency guarantees correctly achieve what they *do* guarantee?
- Can biassing and bounding be combined to model known network failure cases?
- Can transient network disruption be incorporated into this model?

References

1. Thomson, P., Donaldson, A.F., Betts, A.: Concurrency testing using schedule bounding: an empirical study. In: Proceedings of the 19th ACM SIGPLAN symposium on Principles and practice of parallel programming, ACM (2014) 15–28
2. Thomson, P., Donaldson, A.F., Betts, A.: Concurrency testing using controlled schedulers: an empirical study. ACM Trans. Parallel Comput. **2**(4) (February 2016) 23:1–23:37
3. Vinoski, S.: Concurrency with Erlang. IEEE Internet Computing **11**(5) (Sept 2007) 90–93
4. Deligiannis, P., Donaldson, A.F., Ketema, J., Lal, A., Thomson, P.: Asynchronous programming, analysis and testing with state machines. In: 36th Annual ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI'15). (7 2015) 154–164
5. Deligiannis, P., McCutchen, M., Thomson, P., Chen, S., Donaldson, A.F., Erickson, J., Huang, C., Lal, A., Mudduluru, R., Qadeer, S., Schulte, W.: Uncovering bugs in distributed storage systems during testing (not in production!). In: 14th USENIX Conference on File and Storage Technologies (FAST'16). (2 2016) 249–262
6. Zhang, N., Kusano, M., Wang, C.: Dynamic partial order reduction for relaxed memory models. In: Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation. PLDI 2015, ACM (2015) 250–259
7. Musuvathi, M., Qadeer, S.: Iterative context bounding for systematic testing of multithreaded programs. In: Proceedings of the 2007 ACM SIGPLAN Conference on Programming Language Design and Implementation. PLDI '07, ACM (2007) 446–455
8. Musuvathi, M., Qadeer, S.: Fair stateless model checking. In: Proceedings of the 29th ACM SIGPLAN Conference on Programming Language Design and Implementation. PLDI '08, ACM (2008) 362–371
9. Emmi, M., Qadeer, S., Rakamarić, Z.: Delay-bounded scheduling. In: Proceedings of the 38th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages. POPL '11, ACM (2011) 411–422
10. Groce, A., Zhang, C., Eide, E., Chen, Y., Regehr, J.: Swarm testing. In: Proceedings of the 2012 International Symposium on Software Testing and Analysis. ISSTA 2012, New York, NY, USA, ACM (2012) 78–88
11. Marsaglia, G., Tsang, W.W., Wang, J.: Fast generation of discrete random variables. Journal of Statistical Software **11**(1) (2004) 1–11
12. Trangez, N.: An implementation of the raft consensus protocol. https://github.com/NicolasT/kontiki (accessed 2016-10-20).
13. Ongaro, D., Ousterhout, J.: In search of an understandable consensus algorithm. In: 2014 USENIX Annual Technical Conference (USENIX ATC 14), Philadelphia, PA, USENIX Association (June 2014) 305–319

Starling: Stress-Free Concurrency Verification

Matt Windsor[1], Mike Dodds[1], Ben Simner[1], and Matthew J. Parkinson[2]

[1] University of York
{mbw500, mike.dodds, bs829}@york.ac.uk
[2] Microsoft Research Cambridge
mattpark@microsoft.com

Abstract. We introduce *Starling*, a new tool for proving properties of concurrent programs. Starling is based on *program logics*, a form of mathematics for verifying programs in which proofs are close to the program code being verified. This, coupled with its focus on automating as much proof work as possible and a minimalist approach to its construction, make Starling a useful step towards stress-free concurrency verification.

Keywords: Concurrency, verification, automation, program logics.

1 Introduction

Programmers working on operating systems, servers, and other performance-critical scenarios need to tap into the full power of modern CPUs. These CPUs contain multiple *cores*, each capable of simultaneously executing a separate piece of code. Writing programs to work with more than one core is thus a key skill.

The technique of writing programs such that they are able to run on multiple cores at the same time is known as *concurrency*. This is often achieved by dividing the program into multiple separate parts, called *threads*, that run concurrently—perhaps on different cores—and communicate by sharing memory.

Locks and mutual exclusion. Concurrency introduces issues that do not exist in normal programming. Consider, for example, a program with two threads A and B: both are trying to increment a variable x 10 times using the code

```
i = 0; while (i < 10) { x = x + 1; i = i + 1; }
```

If x is 0 initially, one would expect the result to be 20. Yet, any value from 0 to 20 could be left in x after both threads finish. Why? The command `x = x + 1` does three things: it reads x, increments it, and stores back to x. Suppose A reads x and finds it to be 15, but B then performs its increase of x by 1. When A continues, it still thinks x is 15, though it is now 16. It will write $15 + 1$, ie 16, to x, losing B's contribution to x. A has thus *interfered* with B.

This is a common problem in concurrency. One solution is *mutual exclusion*: when one thread has access to x, it forbids others from also having it. We can achieve this using *locks*: algorithms that enforce mutual exclusion by making threads acquire permission to proceed with variable- accessing code before doing so, and relinquish said permission when they no longer need the variable.

As an example of such an algorithm, we consider the *ticket lock*. It works by analogy to a ticket dispenser at, say, a delicatessen:

1. The lock holds a series of tickets labelled $0, 1, \ldots, \infty$. At any stage, it has dispensed n of these tickets: the next one will be labelled n. Of these, k have already taken the lock: the ticket being 'served' is labelled k.
2. A thread wishing to take the lock must first take the ticket, causing n to increase by one. Suppose the ticket it chose was t.
3. The thread must now wait for the lock to serve its ticket. It does so by constantly checking the current value of k, making a note of it (say, as the variable s), and checking whether $s = t$.
4. When $s = t$, the thread acquires the lock.
5. To release the lock, the thread forfeits the ticket, causing k to increase by one: the next ticket may now take the lock.

Proving mutual exclusion for locks. Locks reduce the burden of proving that code is free of unwanted interference, but do not eliminate it. We must prove that the lock satisfies the mutual exclusion property: for locks, this simply means we cannot ever observe the lock being held twice. To do so, we must state said property formally. A first approximation in predicate logic might be

$$holdLock \wedge holdLock \implies false$$

but we have lost the fact that the two *holdLock*s are distinct here.

Finding a valid encoding of mutual exclusion is only the first step towards proving a lock satisfies it. We must show not only that the lock correctly performs the right actions to behave like a lock, but also that the interference of (well-behaved) third parties cannot break the lock. We want a proof technique that

- allows us to specify mutual exclusion in a way that reflects our intuition;
- allows us to reason about interference in a way as simply as possible;
- is easy to use, close to the original code, and automated where possible.

One route forwards is to use a *concurrent program logic*. Based on Hoare's logic for proving properties of non-concurrent programs [3], these combine a formal system for describing shared-memory assertions on shared state with a logical basis for reasoning about whether the actions of a program fulfil such assertions. This line of research has been so fertile that there has been concern [5] over the redundancy in the workload of proving the resulting logics sound. This has led to research on creating logical frameworks, such as the *Concurrent Views Framework* (CVF) [1], to captures the similarities between logics.

We believe that many of the criteria above can be met with relatively little work by turning this process on its head. Instead of using CVF as a base for proving soundness of a large, complex logic, we instead create a small, simple program logic—designed for easy automation—through minimal additions to CVF proper. The rest of this extended abstract discusses the resulting tool.

2 Starling: a tool for proving programs

Starling is a proof tool for a minimal CVF-based program logic. Starling is designed to be fully automatic, to hide (or avoid) as much mathematical complexity as possible, and to be a viable formal proof tool despite this.

Starling is based on the *axiom soundness* rule from CVF: informally, proofs are a series of checks over each command in an program. We must show that each command not only behaves according to assertions made before and after it, but also cannot break any valid assertions the rest of the program may make.

Proving the ticket lock in Starling. We demonstrate Starling with a proof of mutual exclusion for the ticket lock. Starling proofs begin with an *outline* of the program to prove. These have a syntax similar to that of the *C* language, but with some additions. First, the `<command>` notation marks `command` as an *atomic action*: observers cannot see any state between its commencement and its completion. (All actions affecting shared state must be atomic.)

Second, Starling expects a *view*—an assertion of the form {| *A(x) * B(y) * ⋯ * Z(n)* |}—between each command in the outline. Each component of a view, called an *atom*, represents some fact about the shared state that must hold at that point. The *** operator conjoins atoms into views (we discuss how below). Views can contain no atoms (*emp*), or be conditional on some local observation (*if s == t then holdLock() else holdTick(t)*).

We can now write a Starling outline of the ticket lock:

```
shared int n, k; thread int t, s;

method lock() {
  {| emp |}
    <t = n++>; // Fetch and increment ticket in one action.
  {| holdTick(t) |} // By doing the above, we now hold ticket t.
    do {
      {| holdTick(t) |}
        <s = k>;  // Repeatedly fetch and check k against t.
      {| if s == t then holdLock() else holdTick(t) |}
    } while (s != t);
  {| holdLock() |}
}

method unlock() {
  {| holdLock() |}
    <k++>; // Increment
  {| emp |} // We no longer claim we hold a lock.
}
```

We now need to define what these views mean. First, we define the atoms in use:

```
view holdTick(int t);  // we know we hold a ticket with number 't'
view holdLock();       // we know we hold a lock
```

This tells Starling that the alphabet of atoms is *holdTick(t)* where `t` is an integer, and *holdLock()*. Then, we can give views meaning by assigning them to Boolean *constraints*, which filter the possible shared states in which that view can be observed. We can thus realise our earlier intuition about mutual exclusion:

```
constraint holdLock() * holdLock() -> false;  // Mutual exclusion!
```

Starling's power comes from the fact that we can assign any view, or part thereof, a meaning greater than the sum of its parts. This extends to the empty view. We assign a predicate to *emp* to make it invariant (it must *always* hold), then finish by adding more constraints needed for the proof:

```
constraint emp -> n >= k;  // This always holds.
constraint holdTick(a) -> n > a;
constraint holdLock() -> n != k;
constraint holdLock() * holdTick(a) -> k != a;
constraint holdTick(a) * holdTick(b) -> a != b;
```

This concludes our proof of mutual exclusion of the ticket lock, which Starling can now check automatically. This is done through Starling's axiom soundness proof rule, which can be shown to be a strengthening of the CVF rule informally stated earlier. For each command and control-flow transition C in a program, we find the view P immediately before it, and the view Q immediately after. Then, for each `constraint` R `->` $D(R)$, we prove that

$$[\![C]\!] \wedge \lfloor P * (Q \setminus R) \rfloor \implies D(R)$$

where $[\![C]\!]$ takes the semantics of C as a two-state predicate, $\lfloor \_ \rfloor$ maps views to their predicate meanings by collecting the right-hand sides of the applicable `constraint`s, and $\setminus$ subtracts one view from another (ie. an adjoint of $*$).

When all `constraint`s are known, each term above is a simple Boolean predicate and can be checked quickly by *satisfiability-modulo-theories* solvers such as *Z3* [4]. The above proof rule also resembles a Horn clause, with the views as uninterpreted functions. To check a system of such clauses, we need not define the views: a Horn solver such as *HSF* [2] will infer valid definitions if it can. In Starling, we exploit this by allowing `constraint` bodies to be replaced with `?`. This asks Starling to infer those constraints, then try to prove axiom soundness with them. This relieves the proof writer of a large amount of the proof burden.

3 Conclusion

We have demonstrated Starling using the ticket lock. We have also used Starling to prove properties of spinlocks, Peterson's algorithm, and a lock-free version of the multiple-update problem discussed earlier. These additional examples, as well as Starling itself, are available at `https://github.com/septract/starling-tool`.

Future work includes adding support for variable types beyond integers and Booleans, making Starling's frontend more accessible to programmers, and supporting more proof backends. We would also like to formalise the additions and changes we have made to the Views framework. However, as a proof of concept, we feel that Starling is already successful.

References

1. Dinsdale-Young, T., Birkedal, L., Gardner, P., Parkinson, M., Yang, H.: Views: Compositional reasoning for concurrent programs. POPL (2013)
2. Grebenshchikov, S., Lopes, N.P., Popeea, C., Rybalchenko, A.: Synthesizing software verifiers from proof rules. In: Proceedings of the 33rd ACM SIGPLAN Conference on Programming Language Design and Implementation. pp. 405–416. PLDI '12, ACM, New York, NY, USA (2012), http://doi.acm.org/10.1145/2254064.2254112
3. Hoare, C.A.R.: An axiomatic basis for computer programming. Commun. ACM 12(10), 576–580 (Oct 1969)
4. de Moura, L., Bjørner, N.: Z3: An Efficient SMT Solver, pp. 337–340. Springer Berlin Heidelberg, Berlin, Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-78800-3_24
5. Parkinson, M.: The next 700 separation logics. In: Leavens, G., O'Hearn, P., Rajamani, S. (eds.) Verified Software: Theories, Tools, Experiments, Lecture Notes in Computer Science, vol. 6217, pp. 169–182. Springer Berlin Heidelberg (2010)

Part IV

Poster Abstracts

Boundary element modelling of KEMAR for binaural rendering: Mesh production process and validation

Kat Young, Tony Tew, and Gavin Kearney

Department of Electronics
University of York, Heslington, YO10 5DD, UK
kaey500@york.ac.uk

Abstract. Head and torso simulators are used extensively within acoustic research, often in place of human subjects in time-consuming or repetitive experiments. Particularly common is the Knowles Electronics Manikin for Acoustic Research (KEMAR), which has the acoustic auditory properties of an average human head. As an alternative to physical acoustic measurements, the boundary element method (BEM) is widely used to calculate the propagation of sound using computational models of a scenario. Combining this technique with a compatible 3D surface mesh of KEMAR would allow for detailed binaural analysis of speaker distributions and decoder design - without the disadvantages associated with making physical measurements.

This poster details the development and validation of a BEM-compatible mesh model of KEMAR, based on the original computer-aided design (CAD) file and valid up to 20 kHz. Use of the CAD file potentially allows a very close match to be achieved between the mesh and the physical manikin. The mesh is consistent with the original CAD description, both in terms of overall volume and of local topology, and the numerical requirements for BEM compatibility have been met. Computational limitations restrict usage of the mesh in its current state, so simulation accuracy cannot as yet be compared with acoustically measured HRTFs. Future work will address the production of meshes suitable for use in BEM with lower computational requirements, using the process validated in this work.

Keywords: binaural, head-related transfer functions, KEMAR, computational modelling, boundary element modelling

Modelling the Dynamics of Electrooptically Modulated Vertical Compound Cavity Surface Emitting Semiconductor Lasers.

N. F. Albugami, E. A. Avrutin

Department of Electronics, University of York, York YO10, 5DD, UK
na668@york.ac.uk , eugene.avrutin@york.ac.uk

Vertical-cavity surface-emitting laser (VCSEL) constructions capable of direct modulation at bit rates in excess of 40 GBit/s have attracted considerable attention for future high speed long- and medium-haul networks. The two main approaches to realising this goal are, firstly, the improvement in the direct modulation laser performance, with 40 GBit/s direct modulation having been demonstrated recently [1], and, secondly, using modulation of the photon lifetime in the cavity as an alternative to current modulation [2]. Advanced semiconductor lasers involving direct modulation of the photon lifetime promise substantially better dynamic properties than current modulated lasers, because their operating speed is not limited by the electron-photon resonance frequency. Several laser designs to implement this principle have been proposed, and initial measurements and theoretical studies using simple rate equation models [3] [4] are all promising.

Our initial work was on advancing such analysis to include small signal analytical approach. A somewhat unusual result of this analysis is that the concept of 3 dB cutoff frequency as a measure of modulation speed is not applicable to electrooptically coupled-cavity VCSELs as it tends to infinity beyond a certain critical current, their modulation properties are governed instead by the shape and magnitude of the broad resonant peak at the electron-photon resonance frequency.

In order to understand the laser performance more accurately, a model involving careful analysis of both amplitude and frequency/phase of laser emission, as well as the spectrally selective nature of the laser cavity, is required. We have developed such a model, based on the analysis for complex eigenvalues (frequency detunings and amplitude variation rates) of the compound caviy modes and used it to describe the laser operation and predict the performance beyond current experimental conditions, under both small- and large signal modulation conditons. The photon lifetime in the modulator cavity was found to be the ultimate limitation of the modulation speed.

1. S.A. Blokhin, J.A. Lott, A. Mutig, G. Fiol, N.N. Ledentsov, M.V. Maximov, A.M. Nadtochiy, V.A. Shchukin, D. Bimberg, Electronics Letters, 45 (2009) 501-502.
2. V.A. Shchukin, N.N. Ledentsov, Z. Qureshi, J.D. Ingham, R.V. Penty, I.H. White, A.M. Nadtochy, M.V. Maximov, S.A. Blokhin, L.Y. Karachinsky, I.I. Novikov, Applied Physics Letters, 104 (2014) 051125.
3. E.A. Avrutin, V.B. Gorfinkel, S. Luryi, K.A. Shore, Applied Physics Letters, 63 (1993) 2460-2462.
4. V.A. Shchukin, N.N. Ledentsov, J.A. Lott, H. Quast, F. Hopfer, L.Y. Karachinsky, M. Kuntz, P. Moser, A. Mutig, A. Strittmatter, V.P. Kalosha, D. Bimberg, 2008, pp. 68890H-68890H-68815.

Jordan Algebra Artificial Chemistry: Exploiting Mathematical richness for Open-Ended Design

Penelope Faulkner[1,2]

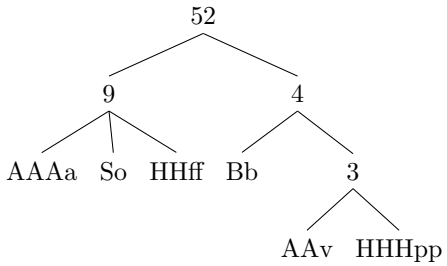
[1] Department of Chemistry,University of York, York, UK
[2] York Centre for Complex Systems Analysis, University of York, York, UK
pf550@york.ac.uk

Abstract. We show through an example the process of linking and decomposing a particle in a Jordan Algebra Artificial Chemistry (AChem). We do this at the level of the tree of the particle.

Keywords: Artificial Chemistry, Artificial Life, Jordan Algebra, Matrices, Linear Algebra, Complexity

1 Poster Abstract

We give a brief introduction to an Artificial Chemistry based on Matrices and Jordan Algebra. To do this we provide walk through an example of forming a particle made up of 6 atoms: AAAa, MMff, Bb, So, AAv and HHHpp. These are combined in four stages by four different links producing the particles 3, 4, 9 and 52. The resultant particle 52 is represented as a tree:



We then also walk through the decomposition of the particle back to the atoms. Through out we also include where space permits the general principals along side the precise application of them. This is based on the work presented in a recent poster [1].

References

1. Faulkner, P., Sebald, A., Stepney, S.: Jordan algebra AChems: Exploiting mathematical richness for open ended design. In press

Evaluation of a Web-App to Support Healthy Living by Older Adults

Zaidatol Haslinda Abdullah Sani[1], Helen Petrie[2]

Department of Computer Science, University of York, United Kingdom

{zas508,helen.petrie}@york.ac.uk

Abstract. Mobile apps are becoming a promising tool to promote healthy attitudes and behaviours among older adults. A preliminary field study was conducted with five older adults of a web-app to support healthy living, in terms of eating fruit and vegetables, and to drinking liquids. Preliminary results show that although there were some usability issues, overall participants found the app easy to use, straightforward and designed as what it supposed to do. The System Usability Scale (SUS) score of 94.5 indicates that the app was highly usable. For most of the participants, the app acted as a motivational tool for them to change their daily diet to eat more fruit and vegetables, and to drink more liquid.

Keywords: mobile apps, older adults, usability, healthy eating, healthy liquid intake, persuasive technology

EINCKM: An Enhanced Prototype-based Method for Clustering Evolving Data Streams in Big Data

Ammar Al Abd Alazeez*, Sabah Jassim, and Hongbo Du

Department of Applied Computing
University of Buckingham, Buckingham, UK
{1405097*, sabah.jassim, hongbo.du}@buckingham.ac.uk

Abstract. The data stream clustering problem is becoming an active research area in big data. It refers to group constantly arriving data records in large chunks to enable dynamic analysis/updating of information patterns/trends conveyed by the existing clusters, the outliers, and the newly arriving data chunk. Examples of such scenarios include web clicks, social network interactions, and global stock market transactions. Prototype-based algorithms for solving the problem have their promises for simplicity and efficiency. However, existing implementations have some limitations in relation to quality of clusters, ability to discover outliers outside clusters, and little consideration of possible new trends/patterns in different chunks. In this paper, a new incremental algorithm called Enhanced Incremental K-Means (EINCKM) is developed. The algorithm is designed to detect additional clusters in incoming data chunks and merge existing outliers to current clusters to generate modified clusters and outliers for the next round. It applies a huristic-based process to estimate number of clusters (K), a radius-based technique to merge overlapped clusters, and a variance-based mechanism to discover the outliers. The algorithm was evaluated on a synthetic dataset using various quality measures. The initial experimental results indicate improved clustering correctness with a comparable time complexity to existing methods dealing with the same kind of problems.

Keywords: Big Data, Data Stream Clustering, Outliers Detection, Prototype-based Approaches

Transformation from Activity Diagram to Class Diagram

Muideen Ajagbe, Fiona A.C Polack and Richard F. Paige,
Department of Computer Science
University of York, York, UK YO10 5DD
Email: maa589@york.ac.uk

Abstract

The paper presents a practical approach to the transformation of UML-style activity diagrams to class diagrams and code. Activity diagrams have been widely used to model behavior; our work focuses on models used in the design of simulations used in research on of complex immune systems in York Computation Immunology Lab. The ultimate goal is to use automated model transformation to improve the reusability and maintainability of such simulators.

Keywords— model, transformation, model-driven engineering, diagram, code.

Social-Insect-Inspired Adaptive Hardware Systems

Matthew Rowlings, Andy M. Tyrrell, and Martin A. Trefzer

Department of Electronics, University of York

Large social insect colonies require a wide range of important tasks to be undertaken to build and maintain the colony. There is a crucial equilibrium between the number of workers performing each task that must not only be maintained but must also continuously adapt to sudden changes in environment and colony need. What is most fascinating is that social insects can sustain this balance without any centralised control and with colony members that have relatively little intelligence when considered individually. Due to this simplicity and evident scalability, it would seem that social insects have evolved a scalable approach to task allocation. The decentralised nature of this model is highly desirable for future high-logic density hardware platforms and will be required as many-core systems start to scale into the hundreds and thousands of nodes, rendering traditional design space exploration approaches infeasible. Many biologists have studied the task allocation of social insects and a comprehensive review of different models is considered in [1]. This gives us a choice of models that we could use for task allocation on the many-core. We investigate the parallel distributed model Gordon proposes in [2] for many-core systems by implementing simple threshold decision functions local to each core, based on "hunger", i.e. idle state, or task affinity due to incoming packages. We have implemented this biological model of task allocation in ant colonies and applied this to a 36-core Network on Chip. We show that effective decentralised task allocation is achieved with very low hardware overhead. We have also shown how a simple behaviour at each node results in a model that intrinsically copes well with faults, allowing autonomous systems to gracefully bring back functionality from even catastrophic failure situations. This has been furthered by introducing more complexity into the intelligence, a panic-like reaction allows approaching deadlines to be serviced by exploiting the task switch to ensure packet deadlines are met. Our results indicate that our approach will be an appropriate mechanism to overcoming the problems introduced by Dark Silicon and other issues of extremely large scale integration that require adaptive solutions.

References

1. S. Beshers and J. Fewell, "Models of division of labor in social insects," *Annual review of entomology*, 2001.
2. D. Gordon, B. Goodwin, and L. Trainor, "A parallel distributed model of the behaviour of ant colonies," *Journal of theoretical Biology*, 1992.

Is Slowness in Parkinson's Disease Unique?

Evolutionary Algorithms in the characterisation of Bradykinesia.

Muhamed SA, Smith SL

Electronics Department, University of York, UK

Abstract. The slowness of movement, clinically termed **Bradykinesia,** is seen in people with Parkinson's Disease (PD) and is the only clinical symptom that is mandatory for diagnosis. Unfortunately, Bradykinesia is also present in other neurological conditions and is difficult to differentiate, especially in the early stages of the disease, leading to a 25% error in diagnosis. It is, therefore important to find out if there are any unique characteristics of Bradykinesia in PD compared to slowness in patients with other conditions. **The main objective of this study is to characterise and differentiate Bradykinesia in PD from other neurological conditions by employing Evolutionary Algorithms (EAs).** An Evolutionary Algorithm is a computational intelligence technique inspired by Darwin's theory of natural selection. Given a problem, an EA will search for a solution by repeatedly selecting the best (or fittest) candidate from a population of possible solutions. EAs have been shown to be capable of evolving classifiers that are highly successful in differentiating types of medical conditions as well as having the potential to inform novel clinical assessment. In this study, subjects with PD and other neurological conditions were asked to undertake three movement tasks used by clinicians to assess Bradykinesia. Movement data was collected using electromagnetic motion tracking sensors that record in six degrees of freedom. Features that indicate Bradykinesia such as movement amplitude, speed and rhythm were extracted and used to train a Cartesian Genetic Program– a type of EA. Receiver operating characteristic (ROC) curves are used to evaluate the performance of the classifier system over a range of thresholds. A further study currently underway will reveal if there are different characteristics of Bradykinesia in PD compared to other neurological conditions such as Huntington Disease and Essential Tremor. It may also reveal underlying characteristics of Bradykinesia towards improving the accuracy of conventional clinical assessment of Parkinson's disease.

Simulations of X-ray Absorption Spectroscopy to study processing effects on Graphene

W. Y. Rojas[1], A. D. Winter[1], S. Kim[2], A. D. Williams[2,6], C. Weiland[3], E. Principe[3], D. A. Fischer[4], D. Prendergast[5], J. Grote[2], and E. M. Campo[1,6]

[1]School of Electrical Engineering, Bangor University Bangor UK LL57 1UT
[2]Materials and Manufacturing Directorate, Air Force Research Laboratory, Wright-Patterson, AFB, Fairborn, OH 45433, USA
[3]Synchrotron Research, Inc. Melbourne, Florida 32901, USA
[4]National Institute of Standards and Technology, Gaithersburg, MD 20899, USA
[5]Molecular Foundry, Lawrence Berkeley National Laboratory, Berkeley, CA 94701, USA
[6]Department of Physics, Wright State University, Dayton, OH 45435, USA
wudmir.rojas@bangor.ac.uk,

Abstract. Nanoelectronics applications based on graphene will owe its effectiveness to processing and large area fabrication. Current processing schemes aim at transferring methods to achieve homogeneous, defect-free, graphene layers onto flexible and non-flexible substrates with the purpose of preserving mobility. Although structural properties of as-grown graphene on Cu and SiC are well documented, further research is required to analyze the resulting lattice structure and strain dynamics (responsible for transport properties) as a consequence of transfer. One approach to assess strain in small molecules is through Near Edge X-Ray Absorption Fine Structure (NEXAFS) spectroscopy. In the experimental realm, studies have confirmed a linear correlation between σ^* resonances and molecular bond lengths, which theoretical models had predicted earlier. In this work, we apply similar arguments to graphene, and use a combination of X-ray spectroscopy experiments and ab-initio simulations to extend the validity of this correlation. This approach has enabled the prediction of average bond lengths and lattice parameters of single and multiple graphenetransferred layers onto a variety of substrates. The eXcited electron and Core-Hole (XCH) approach was followed to simulate NEXAFS spectra based on constrained-occupancy Density Functional Theory. This analysis aims at establishing a correlation between growth and processing, making use of experimental and theoretical data to predict optimum graphitic quality, and ultimately, to inform fabrication routines. The proposed approach suggests NEXAFS spectroscopy, in both its experimental and theoretical modalities, is a powerful technique in the assessment and prediction of structural properties of graphene.

Keywords: Graphene, NEXAFS, Computational physics

PEBBLES: Part of Environment Bio-inspired Blocks for Large-scale Effective Sensing

Connah Johnson[1], Martin Trefzer[2], and Bryce Beukers-Stewart[2]

[1] University of Manchester, Manchester, M13 9PL, Uk.
connah.johnson@student.manchester.ac.uk,
[2] University of York, York, YO10 5DD, UK

Abstract. The PEBBLES (Part of Environment Bio-inspired Blocks for Large-scale Effective Sensing) project proposes the use of biologically inspired algorithms for marine environment monitoring. Ocean acidification combined with rising ocean temperatures apply stresses to marine life. Examples may be drawn from coral bleaching and the overall reduction in health of calcium shelled organisms such as bivalves, an important source of food. As such, an in-situ system monitoring anomalous changes in environment conditions would be of both academic and commercial interest.

Current capabilities for large scale environment monitoring are mainly based on the Internet of Things (IoT) or Big Data methodologies. Implementing IoT technologies on non-well define small systems is non-trivial so is unsuitable for large scale distributed systems. Big Data methods lead to the production of large data sets transmitted to data centers for post-production analysis. The transmittance of raw data leads to security and power management concerns, particularly when considering networks submerged in marine environments. These problems culminate in a system that is unable to make decisions in response to real-time environment perturbations. Interesting consequences center around the sensor system's robustness and adaptability to the local environment. A robust system would recover from sensors lost in hostile environments, overcome online challenges unforeseen in laboratory conditions, and react to stimuli while isolated from human intervention.

The PEBBLES system proposes a different approach to the IoT and Big Data analysis. An intelligent, wireless, distributed sensor network composed of adaptable nodes able to monitor the surrounding environment for anomalous conditions without the need to store or process large data sets. PEBBLES does this through embedding a biologically inspired evolvable algorithm on a nodal basis, encouraging complex emergent properties. This forms a self-healing, self-organising, and adaptive multicellular organism. This work outlines the PEBBLES proposal.

Keywords: adaptive systems, environment monitoring, evolutionary algorithms

Detection and Classification of Retinal Fundus Images Exudates using Region based Multiscale LBP Texture Approach

Mohamed Omar[1], Fouad Khelifi[1], Muhammad Atif Tahir[1]

Department of Computer Science and Digital Technologies, Faculty of Engineering and Environment,
Northumbria University, Newcastle upon Tyne, UK
{mohamed.omar, fouad.khelifi, muhammad.tahir}@northumbria.ac.uk

Abstract. Diabetic retinopathy (DR) is one of the most important cause of vision loss in diabetic patients. The most primary sign of DR is the presence of exudates, and detecting these in early screening is crucial in preventing vision loss. This paper proposes a system for automatic exudate detection using a combination of texture features, extracted from different local binary pattern (LBP) variants, with an artificial neural network (ANN) classifier. The publicly available database DIARETDB0 of color fundus images was used for testing purposes and the values of sensitivity, specificity and accuracy found were 98.68%, 94.81 % and 96.73% respectively for the neural network based classification. These results have also been shown to outperform existing work.

Keywords — Diabetic retinopathy (DR); exudates; fundus image; Local Binary Pattern (LBP); Radial Basis Function; K-Nearest Neighbour; DIARETDB0.

References

1. Shah, A.: Diabetic retinopathy: A Comprehensive Review, Indian Journal of Medical Sciences, vol. 62, no. 12 (2008) 500
2. Omar, M., Hossain, A., Zhang, L., Shum, H.: An Intelligent Mobile-based Automatic Diagnostic System to Identify Retinal Diseases Using Mathematical Morphological Operations (2014) 1–5
3. Akram, M. U., Khalid, S., Khan, S. A.: Identification and Classification of Microaneurysms for Early Detection of Diabetic Retinopathy, Pattern Recognition, vol. 46, no. 1 (2013) 107–116

Modelling Automatic Continuous Emotion Recognition using Hand-Crafted Features and Deep Features using Machine Learning Approach

Yona Falinie A. Gaus, Asim Jan, Jingxin Liu, Fan Zhang,
Rui Qin, and Hongying Meng

Department of Electronic and Computer Engineering, Brunel University London

Abstract. Mental health problem affect one in four citizen at some point of their lives. However, the first step aimed at the behaviour of people suffering from mood disorder only limited to categorize emotion description such as happy, sad, fear, surprise and so on. Our approach is to advance emotion recognition by modelling behavioural cues of human affect as a small number of continuously value time signal, as an emotion felt during present times may be influenced by the emotion during previous one. In the experiments, two types of visual features are extracted from facial expression video and audio, namely hand-crafted features such as audio (LLD low level descriptor), visual (LGBP-TOP and facial landmark), psychological features (ECG,EDA,HRHRV,SCR,SCL) and deep convolutional features. The extracted feature is being used as input to various machine learning approach such as Support Vector Regression (SVR), Random Forest, Linear Regression and so on in which Pearson Correlation Coefficient (CCC) has been used as objective function. For the fusion of multimodal features in decision label, we propose a Weighted-Exponent Decision Fusion based on predication label, to give more weights on well-performed individual features. These methods were tested on the Audio-Visual Emotion recognition Challenge (AVEC2016) and (AVEC2014) dataset, Affect Recognition Sub-Challenge (ASC), and achieves comparable performance when compared to baseline results data.

Secure Key Protection on Commodity Hardware

James A Sutherland

Security Research Group, University of Abertay Dundee, Dundee DD1 1HG

An important challenge in computer security is protecting cryptographic secrets, even in the event that an intruder is able to gain full privilege access (the 'root', Administrator or System account), while still being able to use those keys.

In mobile phones, the SIM card does this, holding the network authentication data securely independently of the handset's own systems. The vast scale – billions of shipments per year – and simplicity made SIM cards a viable solution for that specific task, as well as a vulnerability[**?**]. Expensive dedicated hardware also exists for handling RSA and other keys for the most critical applications, where the expense is justified.

Recent work has attempted to deliver similar security benefits without the financial and performance cost, creating a robustly isolated container using standard processor or chipset facilities.

AMD introduced the SKINIT instruction as part of their secure virtualisation facilities codenamed Pacifica[**?**], which attempted to create a secured minimal container environment through the CPU in conjunction with a TPM; the Flicker project[**?**] built on this to deliver some similar capabilities, but with substantial performance penalties in addition to very specific hardware requirements. As noted in their subsequent paper[**?**], these new CPU facilities were intended to be used only infrequently and so have substantial execution latency. To remedy this and issues with interrupt latency from a long-running section of code with interrupts disabled, the authors propose architectural changes.

By using the older Systems Management Mode instead, this project aims to deliver a versatile facility without the hardware requirements and performance penalties, greatly reducing the impact on interrupt latency by minimising the time spent within the isolated code section.

References

1. Amd pacifica secure virtual machine architecture reference manual. `http://www.mimuw.edu.pl/~vincent/lecture6/sources/amd-pacifica-specification.pdf`. Accessed: 2016-09-19.
2. Sim card database hack gave us and uk spies access to billions of cellphones. `https://www.theguardian.com/us-news/2015/feb/19/nsa-gchq-sim-card-billions-cellphones-hacking`. Accessed: 2016-09-19.
3. Jonathan M McCune, Bryan Parno, Adrian Perrig, Michael K Reiter, and Arvind Seshadri. How low can you go?: recommendations for hardware-supported minimal tcb code execution. *ACM Sigplan Notices*, 43(3):14–25, 2008.
4. Jonathan M McCune, Bryan J Parno, Adrian Perrig, Michael K Reiter, and Hiroshi Isozaki. Flicker: An execution infrastructure for tcb minimization. In *ACM SIGOPS Operating Systems Review*, volume 42, pages 315–328. ACM, 2008.