THE IEEE
Intelligent Informatics
BULLETIN

IEEE COMPUTER SOCIETY

IEEE Computer Society
Technical Committee
on Intelligent Informatics

December 2006 Vol. 7 No. 1 (ISSN 1727-5997)

On-line version: http://www.comp.hkbu.edu.hk/~iib (ISSN 1727-6004)

The Technical Committee on Intelligent Informatics (TCII) of the IEEE Computer Society deals with tools and systems using biologically and linguistically motivated computational paradigms such as artificial neural networks, fuzzy logic, evolutionary optimization, rough sets, data mining, Web intelligence, intelligent agent technology, parallel and distributed information processing, and virtual reality. If you are a member of the IEEE Computer Society, you may join the TCII without cost. Just fill out the form at http://computer.org/tcsignup/.

The IEEE Intelligent Informatics Bulletin

Aims and Scope

The IEEE Intelligent Informatics Bulletin is the official publication of the Technical Committee on Intelligent Informatics (TCII) of the IEEE Computer Society, which is published twice a year in both hardcopies and electronic copies. The contents of the Bulletin include (but may not be limited to):

1) Letters and Communications of the TCII Executive Committee
2) Feature Articles
3) R&D Profiles (R&D organizations, Interview profile on individuals, and projects etc.)
4) Book Reviews
5) News, Reports, and Announcements (TCII sponsored or important/related activities)

Materials suitable for publication at the IEEE Intelligent Informatics Bulletin should be sent directly to the Associate Editors of respective sections.

Technical or survey articles are subject to peer reviews, and their scope may include the theories, methods, tools, techniques, systems, and experiences for/in developing and applying biologically and linguistically motivated computational paradigms, such as artificial neural networks, fuzzy logic, evolutionary optimization, rough sets, and self-organization in the research and application domains, such as data mining, Web intelligence, intelligent agent technology, parallel and distributed information processing, and virtual reality.

Publisher: *The IEEE Computer Society Technical Committee on Intelligent Informatics*
Address: *Department of Computer Science, Hong Kong Baptist University, Kowloon Tong, Hong Kong (Attention: Dr. William K. W. Cheung; Email:william@comp.hkbu.edu.hk)*
ISSN Number: *1727-5997(printed)1727-6004(on-line)*
Abstracting and Indexing: *All the published articles will be submitted to the following on-line search engines and bibliographies databases for indexing—Google(www.google.com), The ResearchIndex(citeseer.nj.nec.com), The Collection of Computer Science Bibliographies (liinwww.ira.uka.de/bibliography/index.html), and **DBLP** Computer Science Bibliography (www.informatik.uni-trier.de/»ley/db/index.html).*

HRL Laboratories: Thinking Outside the Box

APPLIED RESEARCH IN AN AGE OF BOTTOM LINES

I. INTRODUCTION

HRL makes advances in electronics, information & systems sciences, materials, sensors, and photonics: from basic research to product delivery. We are producing pioneering work in high performance integrated circuits, high power lasers, antennas, networking, and smart materials. HRL technologies fly in satellites and fighter jets, ride on diesel locomotives, and support the systems of the future. Each year, HRL's intellectual property base grows with patents and trade secrets in key technology areas.

HRL has a rich history of discoveries and innovations dating back more than 60 years to the days when Howard Hughes first created Hughes Research Laboratories to address the most challenging technical problems of the day. Under that name, and now as HRL, this organization has a long-standing reputation of serving the national interest through contract and internal R&D. We continue to work with government agencies and laboratories, and also collaborate with universities and academic institutions.

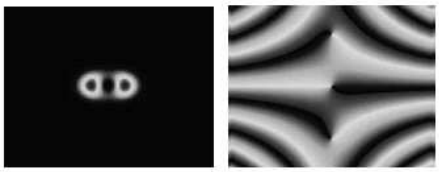


Fig. 1. Amplitude and phase of the wavefunction of two electrons in an anisotropic quantum dot as computed by a few-body code designed by HRL

Over 95% of our our energetic 300-member technical staff have advanced degrees - more than 70% have Ph.D. degrees. We focus on high performance game-changing technologies where we bring unique perspectives and capabilities. Our multi-disciplinary workforce lends itself to development of creative and innovative solutions that cross conventional technology boundaries to produce breakthrough solutions.

This article focuses on the Information and System Sciences Laboratory (ISSL), one of the four technical laboratories at HRL. We will briefly describe the types of research going on there, and then present two representative projects.

II. INFORMATION SCIENCES RESEARCH AGENDA

The ISSL is developing technology to enable smart networks and systems. These are systems that can reason about and adapt to changes in the environment, goals, or their own capabilities, can learn from experience to improve their performance, and can intuitively interact with and respond to their users. This requires broad-based, multi-disciplinary activities in adaptive filtering and learning, human-computer interaction, large-scale networking systems, and computational sciences.

We are combining strengths in mathematics, theoretical physics, computational science and physics-based modeling tools to accurately simulate a variety of important physical phenomena relevant to various experimental groups within HRL (see Figure 1). These models permit realistic analysis of the properties of electronic materials and devices and the phenomena of electromagnetic scattering and propagation.

We apply cognitive science theories to real-world problems, including reasoning by analogy, learning via mental models, and perceiving occluded objects. HRL is actively involved in research on 3D visual and auditory environments, ubiquitous geo-spatial tracking for applications in augmented and virtual reality, and multimodal interaction using dialog and gestures. Applications include command and control, soldier-centric warfare, driver-centric transportation, and remote presence.

In communications and networks, we produced a state-of-the-art wireless platform to analyze and evaluate connectivity, latency, interference, security, quality of services, and congestion issues for a wide variety of application and data networks. Applications include satellite networks, airborne communication networks, vehicular networks, large-scale battlefield networks, and embedded networked sensing systems.

We are developing a single comprehensive architecture to seamlessly integrate perception, memory, planning, decision-making, action, self-learning and affect to address the full range of human cognition. The work focuses on goal-driven scene understanding, language communication, and learning sequentially planned behaviors, as well as on the comprehensive brain-like cognitive architecture.

Fig. 2. A team of pherobots built for Darpa Software for Distributed Robotics program.

We are interested in the dynamics of organization, communication, and control in living organisms, biological systems, and social networks. This is helping us produce high-value systems that exhibit the next-generation capabilities of self-optimization, self-awareness, self-diagnosis, self-regulation, self-healing, self-generation, and reflection.

We are applying evolutionary and neuromorphic techniques to systems where both the software and hardware learn and adapt to their environment. In Fig. 2 we show a swarm of simple robots that coordinate by means of a communications analogue to insect pheromones, to perform mapping of a building and to detect hidden targets.

III. Project Focus on SwarmVision: Advanced Classifiers for Object Recognition and Cognitive Swarms for Fast Search

Objects in a visual scene must be located and classified before they can be combined into events. Typically, classification of objects in an image is performed using features extracted from an analysis window that is scanned across the image. This sequential deterministic search can be very computationally intensive, especially if a small window is used, since a classification must be performed at each window position.

Conventional approaches have utilized motion-based segmentation using background estimation methods to reduce the search space by generating areas of interest around moving objects.This approach fails if the object is motionless or if significant background motion is present, as is the case for motion imagery.

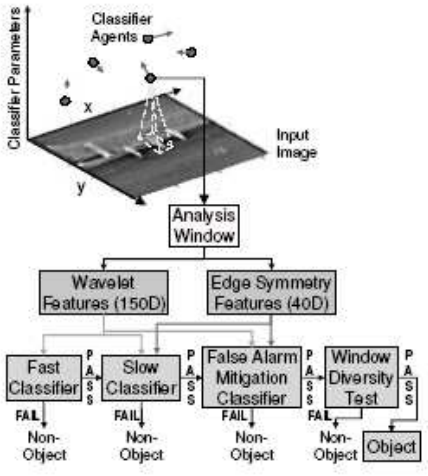


Fig. 3. Cognitive swarms and advanced object classifiers for fast search and object detection in video streams.

HRL's unique cognitive swarm approach to searching for objects combines feature-based object classification with efficient search mechanisms based on the Particle Swarm Optimization (PSO) dynamics developed by Kennedy and Eberhart (1995). Inspired by the flocking behaviors of animals and insects, the PSO algorithm is effective for optimization of a wide range of functions. The algorithm explores a multi-dimensional solution space using a cooperating swarm of search entities or "particles" where the degree of success of each particle in maximizing the objective attracts other members of the swarm. PSO is similar in its generality to genetic algorithms in that it can be used for discontinuous and noisy solution spaces since it only requires an evaluation of the objective function at each particle position; no gradient information or assumptions such as convexity are needed. However, unlike genes in genetic algorithms that compete with each to *win* in a competition for good solutions, in PSO the particles cooperate to explore the solution space and find good solutions. This results in highly efficient search properties. In addition, the evolution of good solutions is stable in PSO (e.g., small changes in the representation result in small changes in the solution), which results in improved convergence compared to GA.

The basic cognitive swarm concept is illustrated in Fig. 3. The objective is to find multiple instances of an object class in an input image. The "cognitive" PSO particles move in a solution space where two of the dimensions represent the x and y coordinates in the video frame. The key concept in our approach is that each particle in the swarm evaluates an objective function value consisting of the classification confidence that the particle's receptive field matches a targeted object in the frame. All cognitive particles in the swarm implement the same classifier, only the classifier parameters vary as the particle visits different positions in the solution space. This recasts the object detection problem as an optimization problem. The solution space dimensions represent location and size of the analysis window and may also include other parameters like rotation.

Cognitive swarms offer a much more efficient method for finding objects in an image compared to searching based on scanning the image, pyramidal approaches, or using gradient information, especially if the scale of the object is not known beforehand. Our experimental results show large speedups over exhaustive search; for example, over 70x speedup to locate and classify one pedestrian of known height (80 pixels) in a 480x700 pixel image.

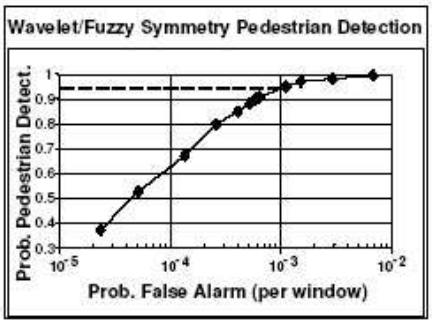


Fig. 4. Window-level probability of detection vs. false alarm rate for human pedestrian classifier that achieves a detection rate of 95% at a very low false alarm rate of 0.1%, and 98% detection for a false alarm rate of 0.3%.

The number of false alarms per image is greatly reduced because the focus of attention of the swarm is quickly directed towards likely objects, which is very important for practical applications (see Fig. 4). The results shown in the figure were obtained on videotaped humans in urban and rural environments under various illumination conditions. The analysis window classification time for our pedestrian classifier is 0.3 msec on a 3 GHz PC. This combination of accuracy and speed is superior to any published results known to us. The framework also provides a natural way to incorporate expectations based on previous recognition results, moving object cues, or externally-supplied rules. For example, if a vehicle has been detected, a human-detection cognitive swarm can be made to focus its attention near the vehicle to "catch" people exiting or entering.

Fig. 3 illustrated some of the object classifiers HRL has developed. This novel approach for object classification utilizes a combination of Haar wavelet and fuzzy edge symmetry features and a cascade of neural network subclassifiers. The features can be calculated quickly using high speed integer arithmetic. A subwindow must be classified as an object by a subclassifier in the cascade in order to proceed to the next

(higher complexity) subclassifier. Non-object subwindows are usually rejected early in the cascade, resulting in high speed without sacrificing accuracy.

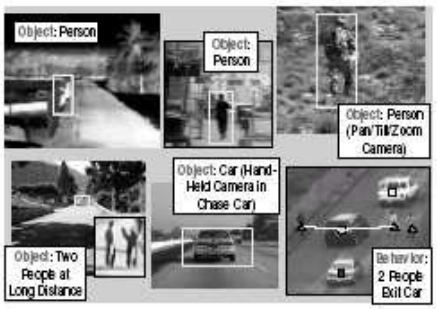


Fig. 5. Detection Examples.

We have used our classifier methodology to create classifiers for other objects as well, such as vehicles and boats. Some example cognitive swarm detections using HRL's advanced object classifiers and cognitive swarms are shown in Fig. 5. HRL has used cognitive swarms successfully in applications for our LLC members, and we are currently adapting them for weapons detection.

IV. Project Focus on System Health Prognosis

Diagnosis of a system determines what failed in the system. It uses observations of the failure such as symptoms of failure, or failed tests. In contrast, prognosis asks what is likely to fail in the near future. It requires not only evidence about present system health as measured by sensors, but also data on health trends, the extent of past system use (e.g., miles, hours, or cycles of operation), and expected future use (possibly focused on a particular mission for which we make prognosis). These multiple pieces of evidence are combined to arrive at system health prognosis.

We have developed a novel framework for prognosis, Fig. 6. The heart of the framework is a probabilistic reasoning engine that produces probability of failure of system components at the end of the mission. It employs a Bayesian network model of the system and multiple sources of evidence for prognosis. The evidence about the previous usage and expected usage for the mission, i.e. future usage, is derived from maintenance and usage data bases and from mission specification. The evidence about present health of components is obtained by applying signal processing and feature extraction algorithms on sensor measurements. Health history of components is used to project health into the future i.e. to the end of the mission. Here trending algorithms are applied to produce the evidence. All elements of the evidence are fused in the reasoner.

Bayesian networks were first proposed as a tool for reasoning in the presence of uncertainty nearly twenty years ago. Many diagnostic systems based on Bayesian networks have been described in literature and some of them have been implemented and deployed in the field. But application of Bayesian networks to prognosis requires a reasoner that is different from those used for diagnosis. An example of a Bayesian network graph developed for a flight actuator is shown in Fig. 7. The graph constitutes a structure of the model. The nodes of the graph are annotated with model parameters, which are conditional probability tables. In Fig. 7 they are shown as histograms.

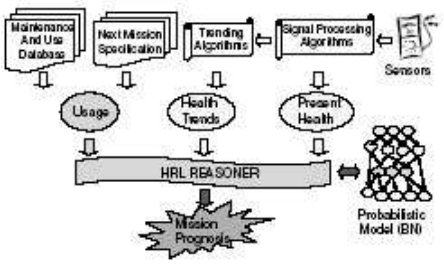


Fig. 6. Prognosis Framework Based on Bayesian Network Models and Probabilistic Reasoner.

In two phases our novel reasoner supports both diagnosis and prognosis. In phase one, a diagnostic phase, the inputs are the evidence on the present system usage and the present health. Given this evidence and system model, the reasoner produces a list of component failures ranked by probability of occurrence. In phase two - prognosis phase - the reasoner takes in evidence on usage for the future mission and evidence on health trends at the end of the mission. The output is a ranked list of probabilities of component failures at the end of the mission.

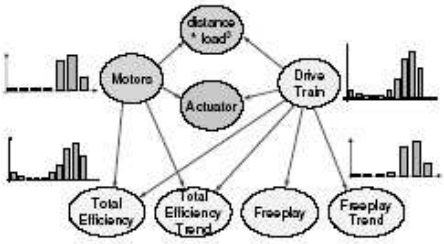


Fig. 7. Bayesian Network Model for Flight Actuator - Structure and Distributions. Motors and Drive Train represent components, Actuator is a subsystem, the node at the top stands for evidence of usage and the four nodes along the bottom represent present and future health evidence.

In addition to the unique reasoner we have also designed a special layered form of Bayesian network. The structure and parameters of the network are customized to diagnosis and prognosis. The layered Bayesian model is much easier to create and requires fewer parameters. Moreover it reduces the computational burden during reasoning. We have developed an editor for the layered Bayesian models, which uses simple tabular representation of the model information. It is intended for experts familiar with the system and does not require knowledge of Bayesian networks. We have also developed a family of software tools for diagnostic/prognostic model evaluation and debugging.

We have used our methodology and tools in development of diagnosis and prognosis solutions for many real-life complex systems including diesel locomotives, automobiles, and aircraft. Our solutions became a part of commercial software provided for some of the systems. We were also successful in extending our methodology and tools to other problems such as decision support for law enforcement and data analysis for homeland security related purposes.

Contact Information
President: Dr. M.W. (Matt) Ganz
310-317-5200
mganz@hrl.com

VP: Dr. C.G. (Conilee) Kirkpatrick
310-317-5374
ckirkpatrick@hrl.com

HRL Laboratories, LLC
3011 Malibu Canyon Rd.
Malibu, CA 90265
http://www.hrl.com

Supporting Provenance in Service-oriented Computing Using the Semantic Web Technologies

Liming Chen and Zhuoan Jiao

***Abstract*—The Web is evolving from a global information space to a collaborative problem solving environment in which services (resources) are dynamically discovered and composed into workflows for problem solving, and later disbanded. This gives rise to an increasing demand for provenance, which enables users to trace how a particular result has been arrived at by identifying the resources, configurations and execution settings. In this paper we analyse the nature of service-oriented computing and define a new conception called augmented provenance. Augmented provenance enhances conventional provenance data with extensive metadata and semantics, thus enabling large scale resource sharing and deep reuse. A Semantic Web Service (SWS) based, hybrid approach is proposed for the creation and management of augmented provenance in which semantic annotation is used to generate semantic provenance data and the database management system is used for execution data management. We present a general architecture for the approach and discuss mechanisms for modeling, capturing, recording and querying augmented provenance data. The approach has been applied to a real world application in which tools and GUIs are developed to facilitate provenance management and exploitation.**

I. Introduction

PROVENANCE is defined, in the Oxford English Dictionary, as (i) the fact of coming from some particular source, origin, derivation; (ii) the history or pedigree of a work of art, manuscript, rare book, etc. This definition regards provenance as the derivation from a particular source to a specific state of an item, which particularly refers to physical objects. For example, in museum and archive management, a collection is required to have archival history regarding its acquisition, ownership and custody.

Provenance is an important requirement in many practical fields. For instance, the American Food and Drug Administration requires that the record of a drug's discovery be kept as long as the drug is in use. In aerospace engineering, simulation records that lead up to the design of an aircraft are required to be kept up to 99 years after the design is completed. In museum and archive management a collection is required to have archival history regarding its acquisition, ownership and custody.

In computer-based information systems, research on provenance has traditionally been undertaken in the arena of database systems under different banners such as audit trail, lineage, dataset dependence and execution trace [2] [3]. For example, the Chimera Virtual Data System [4] addresses data linage with the Chimera virtual data schema. Similar works were also described in [5] [6]. The common feature of these systems is that they try to trace the movement of data between data sources and obtain information on the "where" and "why" of a data item of interest as a result of a database operation. A separate thread of research, i.e. the so-called knowledge provenance, concentrated on explaining information provenance for Web applications [7] [8]. The research placed special emphasis on source meta-information and knowledge process information, in particular, the reasoning process used to generate the answer.

Recently, research on data provenance in service oriented computing has received growing attention [9] [10] [11] as the enabling Web/Grid service technologies and the infrastructure for Service Oriented Architecture (SOA), such as the Open Grid Service Architecture (OGSA), become mature and available. In a SOA, resources on the Web/Grid, including hardware, software code, application systems and knowledge, are regarded as services; and such services are brought together to solve a given problem typically via a workflow that specifies their composition. The running of an application programmed in a SOA style requires the enactment and execution of the workflow, which is referred to as a process. Web/Grid services are dynamic and distributed in nature, i.e. they can be published and withdrawn to/from the Web/Grid arbitrarily. This means a solution (a workflow) to a problem may not be always available or consists of the same set of services at different time of problem solving. Thus, recording and archiving how a result is derived becomes critical in order to validate, repeat and analyse the obtained results.

Data provenance in a SOA/OGSA is concerned with the entire execution history of a service workflow that leads to the particular result, i.e. evolving from traditional "data-centered" provenance towards "process-centered" provenance. An initial attempt has been made in myGrid project (www.mygrid.org.uk) where log files have been annotated and recorded for experiment validation and recreation [12]. A systematic research is conducted in the EU PROVENANCE project

Liming Chen is with the School of Computing and Mathematics, University of Ulster, Co. Antrim BT37 0QB, U.K. (e-mail: l.chen@ulster.ac.uk).

Zhuoan Jiao is with School of Engineering Sciences, University of Southampton, Southampton SO17 1BJ, UK. (e-mail: z.jiao@soton.ac.uk).

(twiki.gridprovenance.org/bin/view/Provenance) aiming to develop a generic architecture for capturing, recording and reasoning provenance data [13]. The project also intends to propose protocols and standards to formally standardize *provenance computing* in SOA/OGSA.

At the time of writing, most provenance systems focus on capturing and recording execution data passed between services within a workflow. Metadata about services, such as the quality of services, their parameters (functional and non-functional), and workflows are scarce and informal. There are no formal representation and common semantics. This imposes severe limitations on the interoperability, searchability, automatic processing capability and reasoning of provenance data, and ultimately the use and reuse of services.

This paper aims to tackle the aforementioned problems by exploiting the Semantic Web technologies, and our research contributions are: (1) introducing the conception of augmented provenance based on the characteristics of service-oriented computing, which enhances conventional provenance with rich metadata and formal semantics; (2) proposing a Semantic Web Service (SWS) based hybrid approach to supporting augmented provenance; (3) desiging and prototype implementing a system architecture for the proposed approach. Our work is motivated by the realisation that SOA/OGSA-based applications require extensive rich metadata in multiple facets, at multiple levels of granularities in order to make effective use of previous problem solving expertise. The central idea of the approach is to capture provenance data from the semantic descriptions of the web services, thus enabling the use of the Semantic Web technologies for provenance data representation and storage. We place special emphasis on semantics, particularly the ontological relationships among diverse metadata, which enables deep use of provenance by reasoning.

The remainder of the paper is organized as follows: Section 2 analyzes the characteristics of service-oriented computing from which we draw the conception of augmented provenance. Section 3 describes the proposed approach and its system architecture for managing augmented provenance. We give an application example in Section 4 and discuss our experiences and lessons in Section 5. Section 6 concludes the paper and points out some future work.

II. AUGMENTED PROVENANCE FOR SERVICE-ORIENTED COMPUTING

We have defined the concept of *augmented provenance*, after analyzing the key characteristics of provenance data in a SOA. We believe this is more instructive than trying to produce an all embracing conceptual definition. To help clarify our conception of augmented provenance and justify our proposed approach, we present below a motivating scenario that captures what we believe are the requirements of provenance in a SOA/OGSA..

A. A motivating scenario

This scenario is based on the UK e-Science project *Grid-enabled Optimisation and Design Search in Engineering* (GEODISE). Engineering Design Search and Optimisation (EDSO) is a computationally and data intensive process whereby existing engineering modeling and analysis capabilities are exploited to yield improved designs. An EDSO process usually comprises many different tasks. Consider the design optimization of a typical aero-engine or wing, it is necessary to (1) specify the wing geometry in a parametric form, (2) generate a mesh for the design, (3) decide which analysis code to use and carry out the analysis, (4) decide the optimisation schedule, and finally (5) execute the optimisation run coupled to the analysis code. Apparently a problem solving process in EDSO is a process of constructing and executing a workflow.

GEODISE aims to aid engineers in the EDSO process by providing a range of Internet-accessible Web/Grid services comprising a suite of design optimization and search tools, computation packages, data management, analysis and knowledge resources. In the GEODISE problem solving environment services are composed into a workflow which is subsequently enacted and executed. The executed workflow is described by a XML file which is stored in the database together with limited metadata such as the file's size, location, etc [14].

After the system was introduced to engineers, a number of questions have been raised regarding to the service and workflow reuse. For instance, engineers may want to find a workflow that uses a particular service *S1*; to find workflows that use a service with the similar algorithm to the algorithm used by *S1*, or to find a similar service to replace service *S1* used in the current workflow and re-run the workflow. To answer these questions, we identify a number of requirements for provenance data, as described below.

Firstly, provenance should include metadata at multiple levels of abstraction, namely process level, service level and parameter level. For example, a workflow instance with all its parameter settings and values is a provenance record for the data derived from it, but the workflow itself also needs provenance information, i.e. which workflow specification was it instantiated from, who enacted it, etc.

Secondly, provenance should include metadata in multiple facets. These may include knowledge provenance, e.g. what knowledge is involved and used; and the decision provenance, e.g. how a decision was arrived at, etc. Each facet of provenance has its roles and uses, and different applications have different emphases and requirements for provenance.

Finally, provenance is not only used to validate, repeat and analyze previous executions but, more importantly, to further advance investigation and exploration based on the previous results. In EDSO an optimisation can be performed using different services (algorithms), and each of them can generate different qualities of results. Engineers, particular novices, usually start a new design by looking at previous best design practices (workflows), and perform design search and optimization by changing constituent services and/or tuning control parameters of the previous workflows. This requires knowledge and decision trails become an indispensable part of the provenance.

B. *Provenance analysis and augmented provenance*

The essence of service-oriented computing is the sharing and reuse of distributed, heterogeneous resources for coordinated problem solving in dynamic, multi-institutional virtual organizations (VO). Service-oriented computing has the characteristics of dynamic service provisioning and across-institutional sharing, i.e. VOs are formed or disbanded on-demand. In such environments a workflow consists of services from multiple organizations in a dynamic VO. The success of workflow execution depends on domain knowledge for service selection and configuration, and a mutual understanding of service functionalities and execution between the service providers and consumers. The complexity of a problem solving process requires not only the execution data of a workflow (e.g. the inputs and outputs of services, the configuration of service control parameters), but also rich metadata about the services themselves (e.g. their usages, the runtime environment setting, etc.), in order to validate, repeat and further investigate the problem solving process at a later stage.

While specific domains or applications determine the actual levels of abstraction and interested facets of provenance, we can identify some common characteristics of provenance data in a SOA. First, SOA oriented provenance data contain both execution data and execution independent metadata. The metadata are centered on the key SOA entities, namely workflows, services and parameters.

Second, rich relationships exist among multiple levels and facets of metadata in SOA/OGSA applications. For instance, a workflow consists of services that in turn contain various parameters. Furthermore, services within a workflow, as well as the parameters of a service, may be organized in various ways. The relationships actually form a kind of knowledge model, which can be used to encode domain knowledge. Appropriate modeling of the metadata can facilitate the data retrieval and the discovery of new knowledge through reasoning. For example, a hierarchical tree structure could be used to model the "is part of" relation between workflows, services and parameters; ontological links could be used to denote semantic relations between services, parameters and commonly accepted types.

Third, not all provenance data can be captured automatically, especially those pertaining to knowledge and decision provenance. Annotation and commenting are therefore an important aspect of provenance. For example, in an EDSO experiment, engineers may annotate why a specific service or algorithm or a value for a parameter is selected. They may wish to annotate the performance of a particular service or the quality of overall results so that future designs can be improved based on the annotations.

Text comments and tagging have been traditionally used to add metadata, but they suffer limitations such as the lack of interoperability, the inability of automation, etc. It is obvious that formal modeling and representation of provenance data with explicit semantics are required in order to facilitate automatic, seamless access and sharing of the provenance data.

To differentiate from traditional provenance understanding, we introduce the concept of *augmented provenance*, defined as: the augmented provenance of a piece of data is the process that leads to the data, and the related semantic metadata of the process.

Although our motivating scenario and analysis are based on EDSO, it is not intended to be domain-specific. The scenario depicts the general features of and requirements for provenance in service-oriented computing. Therefore, the augmented provenance conception and the proposed SWS-based approach are broadly applicable to a range of service-based applications.

III. A SWS-BASED HYBRID ARCHITECTURE FOR AUGMENTED PROVENANCE

We propose a SWS-based hybrid architecture for creating and managing augmented provenance as shown in Figure 1. Central to the architecture is the use of SWSs for managing execution-independent metadata and a hybrid mechanism for handling the execution data. The architecture consists of a set of components, namely the Web/Grid Services (WGS), Semantic Web Service Repositories (SWSR), Workflow Construction Environment (WCE), Workflow Enactment Engine (WEE) and Augmented Provenance Management Services (APMS). These components communicate and interact with each other to enable effective and efficient management of augmented provenance, which we discuss in the rest of this section.

A. *A SWS-based perspectives*

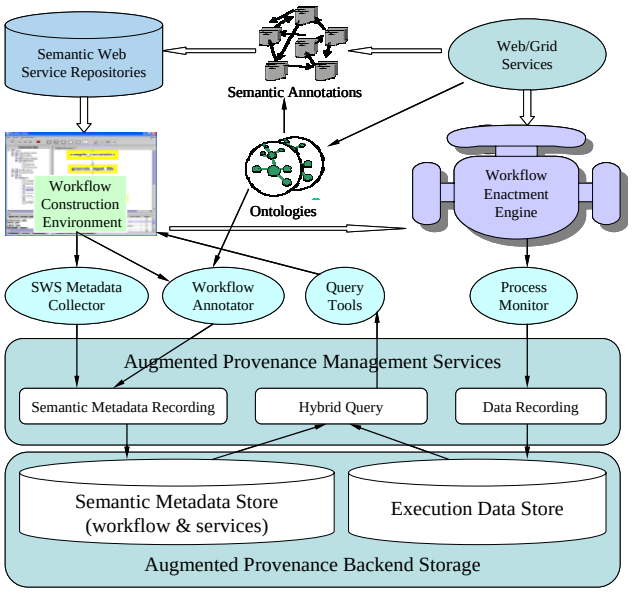


Fig. 1. The augmented provenance architecture

In service-oriented computing, distributed Internet-accessible services such as those contained in the WGS component, serve as the basic computing blocks in SOA/OGSA. As Web/Grid services are described in WSDL [1], published in UDDI (www.uddi.org) and invoked by SOAP, all these technologies provide limited support for service metadata and semantics,

[1] WSDL along with SOAP, RDF, and OWL are W3C standards, please refer to www.w3.org.

thus unable to produce directly augmented provenance.The SWS-based approach uses ontologies and semantic annotation for the acquisition, modeling, representation and reuse of provenance data. The rationales behind this approach are that (1) ontologies can model both provenance data and their contexts in an unambiguous way; (2) provenance data generated via semantic annotation are accessible, shareable and machine processable in a SOA/OGSA; and (3) the Semantic Web technologies and infrastructure can be exploited to facilitate provenance data acquisition, representation, storage and reasoning. More specifically, it will make use of semantic descriptions of Semantic Web Services to generate augmented provenance directly and other Semantic Web technologies such as ontology languages, semantic repository and reasoning for provenance data representation, storage and querying.

The foundation of the architecture is the SWSR component, which contains semantic descriptions of Web/Grid services. SWSR is based on SWS technology that complements current web service standards by providing a conceptual model and language for semantic markup. While the original goal of SWS is to enable the (total or partial) automation of service discovery, selection, composition, mediation, execution and monitoring in service computing, SWS does provide a mechanism for incorporating rich metadata, which can be utilised for provenance purpose. More concretely, SWSR consists of semantically enriched metadata describing the properties and capabilities of services in unambiguous, computer-interpretable form, which can serve as a source of a data item's augmented provenance.

The key enabling technology for SWS is service ontology that provides machine processable models of concepts, their interrelationships and constraints. Service ontology can be used to capture the background knowledge and vocabulary of a domain. For example, OWL-S (www.daml.org/services/owl-s) service ontology defines a number of terms and relationships to describe a service metadata. As an upper service ontology, OWL-S can be further extended based on domain characteristics and application requirements to accommodate domain-specific service description requirements. Semantic descriptions in SWSR are generated by applying service ontologies to services through an annotation tool provided by the Ontological Annotation component. SWSR provides the WCE with a pool of semantically described services through which the WCE can discover and select required services.

Critical to the success of our approach is the WCE component, which collects semantic metadata and records them in provenance stores. WCE allows users to discover and select required services from SWSR locally or on the Web/Grid to compose a service workflow for a given problem. The generated workflow will be passed onto WEE for binding and enactment.

With regards to the provenance, WCE can play three roles, i.e. extracting semantic metadata from service descriptions, generating workflow semantic metadata as part of augmented provenance and performing provenance queries. As WCE uses services from SWSR, the collection of selected services' metadata is straightforward. Each time a service is added into a workflow, the SWS Metadata Collector will retrieve the service's semantic metadata from SWSR and linked to the workflow. For a new workflow, semantic metadata has to be created on the fly because they do not exist in prior.

The Workflow Annotator component will operate in WCE and enable users to describe a workflow in terms of workflow ontology. Workflow's metadata could include a workflow identifier, its creator (i.e. individual or organization), problem solved, date, etc. In practice, an ontology-driven form can be generated automatically from the workflow ontology to help users capture relevant metadata. Some information may be collected directly from the workflow construction process such as date, time, and machine identifiers. Both workflow and service semantic metadata will be submitted to APMS for recording, and later be queried using the Query Tools.

Augmented provenance management services (APMS) are designed for managing augmented provenance data beyond the lifetime of a SOA/OGSA application. It provides recording (archiving) and querying interfaces for augmented provenance backend storage as well as additional administration functionalities such as authentication, authorization and housekeeping. In the context of a SOA/OGSA, provenance backend storage can be decentralized in multiple sites, and APMS are implemented as web services, thus facilitate web accessibility to provenance data and improve the scalability..

B. A hybrid mechanism

Augmented provenance contains execution data generated at the run-time, e.g. the values of inputs and outputs of services; as well as semantic metadata at the design time, e.g. the descriptive information about the workflows, services and parameters. The different nature of these two types of provenance data is reflected in the way they are captured, modeled, represented and stored. To support the heterogeneity of provenance data in a SOA/OGSA, a hybrid approach is adopted, i.e., the approach uses the Semantic Web technologies to handle a workflow's semantic metadata, and the database technologies to deal with execution-dependent process data, thus avoiding duplication and making maximum use of existing DBMS infrastructure. It also proposes a hybrid storage and retrieval mechanism to facilitate coordinated archiving and query of augmented provenance data.

The WEE is responsible for interpreting workflow scripts, binding individual constituent services with corresponding inputs, and invoking executions. A Process Monitor operating in the WEE will extract initial default or user-configured input variable names and values from the interpretation of a workflow script. It will then monitor the execution process of the workflow by querying the execution data repository periodically, thus intermediate and final output results from the workflow's execution could be captured.

As can be seen from the architecture, semantic metadata are collected from WCE and recorded to APMS's Semantic Metadata Store (SMS) via the Semantic Metadata Recording interface. Semantic metadata shall be represented in semantic web languages such as RDF or OWL. Semantic metadata backend store could be a semantic repositories such as 3Store [15] or instance store [16]. Normal workflow execution data will be collected from the WEE and recorded into APMS's Execution Data Store (EDS) via the Data Recording interface.

The execution data backend store could be any commercial database systems.

The APMS operates as follows: each time a workflow is built in WCE, the WCE will store a workflow template in SMS. This template will contain the overall semantic descriptions about the workflow; the semantic metadata for each of the constituent services, including each service's profile metadata and input/output metadata, and an auto-generated unique workflow template ID (UUID, Universally Unique Identifier, www.ietf.org/rfc/rfc4122.txt) as a handle for later reference. An executable workflow based on the workflow template is instantiated by providing values for the required input parameters, and the WEE will store the workflow instance in EDS and associate it with the workflow template ID. If a user reuses a previous workflow template to perform another run without changing the services and the sequence of service execution, the WCE will not record a new workflow template but the WEE will record another workflow instance under the same workflow template ID.

Based on the hybrid storage mechanism, querying augmented provenance data becomes flexible and efficient. A user can use ontologies to frame semantic queries, e.g. in terms of a service profile metadata or a workflow's metadata or a parameter's metadata or any combination of them. Once a workflow template is discovered, all its execution instances can be found from EDS based on the workflow template ID. Further search can be performed to find the set of executed workflows matching other search criteria (e.g. its creator, creation-date, input parameter-values, etc) using the database query mechanism.

The separation of semantic metadata and execution data has many advantages: Firstly, metadata can be formally modeled using ontologies and represented using expressive web ontology languages. This helps capture domain knowledge and enhance interoperability. Secondly, workflow execution usually produces large amount of data that have little added value for reasoning, and the traditional database systems are optimal for handling them. Finally, the hybrid query mechanism provides flexibility and alternatives – users can perform semantics based query or direct database query or a combination to meet application needs.

IV. Application Example

The proposed approach has been applied in GEODISE to manage augmented provenance for grid-enabled service-based EDSO, and in turn the provenance data are used to aid engineers in the design process by answering provenance-related questions. Figure 2 shows the provenance management system in GEODISE, which is described in detail below.

A. Creating semantic metadata

To manage augmented provenance in GEODISE we have built a number of EDSO ontologies, including domain ontology and service ontology, through extensive knowledge acquisition and modelling [17]. Figure 3 shows a fragment of the service ontology developed using Protégé OWL plugin (protege.stanford.edu/plugins/owl). The left column displays ontological concepts while the right column lists ontological properties. We regard a workflow as a composite service.

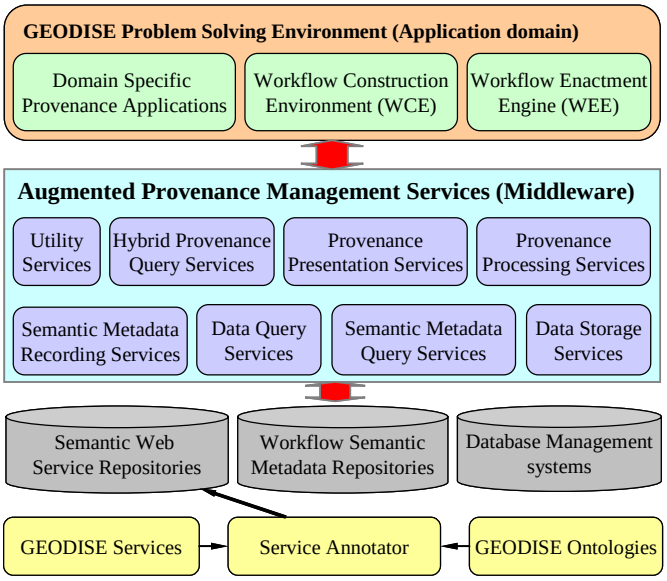


Fig. 2. The provenance management system

Therefore, the service ontology can be used to model semantic metadata for both services and workflows. EDSO service ontology is based on OWL-S upper service ontology. It further extends OWL-S to incorporate EDSO specific metadata such as algorithmUsed, dataPhysicalMeaning, dataUnitType, previousService, followingService, derivedFrom, etc.

We have developed semantic metadata annotation interfaces for capturing semantic metadata. A front-end GUI, known as Service Annotator [19], was developed to help users extract automatically service's metadata, which are then enriched using EDSO domain and service ontologies. The annotation API is also used to implement the Workflow Annotator wizard in WCE to capture and annotate workflow metadata during workflow construction process. The generated semantic metadata for both services and workflows are represented in OWL and stored in the Semantic Web Service Repositories and

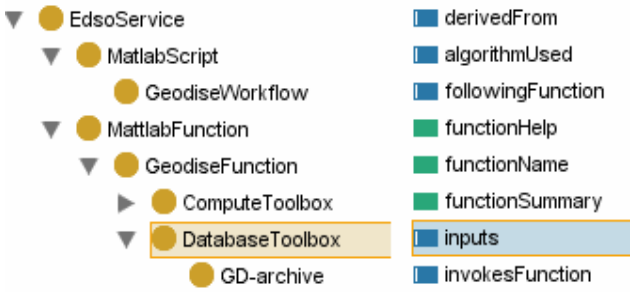


Fig. 3. GEODISE service ontology

Workflow Semantic Metadata Repositories respectively. Both repositories were implemented using the Instance Store technology [16], which provides recording and query interfaces for manipulating semantic metadata. The interfaces use the description logic based reasoning engine Racer [18] to reason over semantic metadata [19].

B. *Collecting and recording execution data*

GEODISE uses Matlab (www.mathworks.com) as its workflow enactment and execution engine. Therefore, input and output variables and their values can be captured and collected from Matlab workspace memory. Acquired execution data are managed by the GEODISE database toolbox [14]. The database toolbox exposes its data management capabilities to the client applications through Java API, as well as a set of Matlab functions. The Java API has been used by the workflow construction environment to archive, query, and retrieve the workflow instances for reuse and sharing; and the Matlab function interfaces allow Matlab scripts to archive, query and retrieve data on the fly at the workflow execution time. Data related to a workflow instance are logically grouped together using the datagroup concept supported by the database toolbox.

C. *Querying augmented provenance data*

Augmented provenance contains rich metadata and semantic relations, which enable users to perform extensive manipulation of provenance data (instead of simple retrieval of data). Such manipulation could include, among other things,

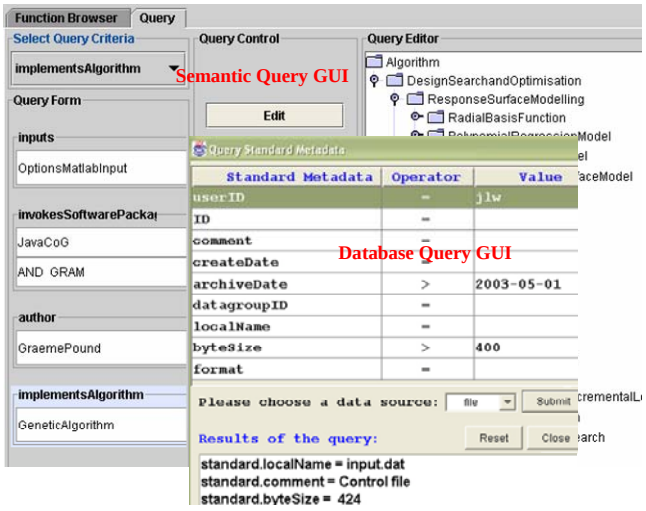


Fig. 4. The query GUIs

retrieving, matching, aggregating, filtering, deriving, inferring and reasoning provenance data in terms of ontological links. This gives rise to many choices and possibilities regarding resource reuse and provenance in addition to validation, repetition and verification. For example, a service of a workflow could be replaced by a semantically compatible service based on augmented provenance.

As an initial step, we have implemented two front-end query GUIs, see Figure 4, to provide dual query mechanisms for flexible and efficient provenance data search and retrieval. The semantic query GUI (i.e. the form) aims to get the high-level provenance data of different facets based on ontology-driven query criteria. The GUI is generated automatically from the EDSO service ontology, and query expression is constructed with support of ontological relations among a workflow, services and parameters. The database query GUI is based on the database schema and can perform keyword-based search and retrieval.

In terms of specific requirements of an application, a user can choose either query GUI accordingly. For example, if a user just wants to know the generic metadata about a workflow profile, its constituent services and types of parameter rather than concrete execution input/output values, a semantic query suffices. To retrieve the full augmented provenance, i.e. both semantic metadata and execution data, a joint query can be launched from either GUI. A workflow's semantic metadata and execution data is cross-referenced using workflow ID.

D. *Provenance services*

To manage augmented provenance, recording interfaces and APIs are needed to accumulate provenance data. A provenance store is not just a sink for provenance data: it must also support some query facility that allows, in its simplest form, browsing of its contents and, in its more complex form, search, analysis and reasoning over process documentation so as to support use cases. Therefore, query interfaces and APIs are an indispensable component in the architecture. Since provenance stores need to be configured and managed, an appropriate management interface is also required.

Apart from the aforementioned fundamental functionality, high-level processing and presentation user interfaces may be required to provide feature-rich functionality. For instance, processing services can offer auditing facilities, can analyse quality of service based on previous execution, can compare the processes used to produce several data items, can verify that a given execution was semantically valid, can identify points in the execution where results are no longer up-to-date in order to resume execution from these points, can re-construct a workflow from an execution trace, or can generate a textual description of an execution. Presentation user interfaces can, for instance, offer browsing facilities over provenance stores, visualise differences in different executions, illustrate execution from a more user-oriented viewpoint, visualise the performance of execution, and be used to construct provenance-based workflows. However, such interfaces typically are application specific and therefore cannot be characterised in a generic provenance architecture.

While interfaces could be implemented in different ways in view of application characteristics and use scenarios, in our example we have provided Web service interfaces for these basic provenance management interfaces. Figure 2 shows the proposed and partially implemented provenance services as system middleware upon which higher-level provenance system or provenance aware applications can be built.

In the system, the recording and query services are responsible for archiving and retrieving augmented provenance data. The Utility Services provide administration facilities such as authentication, authorisation and the lifetime management of provenance data. The processing services provide added-value to the query interfaces by further searching, analysing and reasoning over recorded provenance data. For instance, they can offer such facilities as auditing, comparison of different processes, and check up of semantic consistency and so on. Provenance presentation services provide mechanisms to present query results and processing services' outputs, they are prone to be application dependant. For instance, presentation services can offer browsing, navigation, visualization,

graphical illustration, etc. for provenance data and execution processes. At the time of writing we have developed recording and query APIs and wrapped them into core services, which underpin the implementation of Service/Workflow Annotator and the two query GUIs.

E. Provenance use cases in GEODISE

GEODISE augmented provenance management system enables a number of provenance use cases, some of them are described below.

1) Find the data derivation pathway for a given design result. A user first performs a direct query over the database to retrieve the instantiated workflow description and scripts for the result. Associated input data and generated output data can also be retrieved via the datagroup ID. This workflow script can be enacted in an enactment engine, i.e. Matlab environment, for a re-run.

2) Find information about the optimisation service in the workflow that generates the given result. From the above query, a user can get the workflow template ID through which users can find all involving services, and select the optimisation service to retrieve its associated metadata.

3) Find the similar optimisation algorithms to the one used in this workflow that produces the given result. Following the above query steps we can obtain metadata of an optimisation service, which will contain the type of the optimisation algorithm, e.g. a genetic algorithm (GA). Using the type information in conjunction with the service ontology we can then find out all optimisation services from the SWSR by performing a query based on the algorithmUsed property metadata of the service ontology.

Many other data and/or semantic queries can be framed. For example, find all instantiated workflows that are executed after a specific date; find all workflows that are built by the author who produce this design result.

V. DISCUSSIONS

Whilst provenance has been investigated in other contexts [9] [10] [11], our work concentrates on provenance related to service workflow in a SOA/OGSA. This process-centered view of provenance is motivated by the fact that most scientific and business activities are accomplished by a sequence of actions performed by multiple participants. The recently emerging service-oriented computing paradigm, in which problem solving amounts to composing services into a workflow, is a further motivating factor towards adopting this view.

We identify that augmented provenance in a SOA/OGSA consists of two types of provenance data: execution independent metadata and execution data. We have placed special emphasis on execution independent metadata as Web/Grid services are dynamically published, discovered, aggregated, configured, executed and disbanded in a virtual organisation. Further examination on the motivating scenario shows that execution independent metadata exist at multiple levels of abstraction and multiple facets, and rich relationships exist among them. If such rich metadata can be modeled and represented in a way that semantics and domain knowledge are captured and preserved, it will provide great flexibility and potential for deep processing of provenance data later. This leads to the conception of augmented provenance and further our decisions to use ontologies for metadata modeling and use SWS for capturing semantic metadata.

The employment of service-oriented paradigm for provenance management system is based on several considerations. Firstly, provenance can provide maximum added value for complex distributed applications that are increasingly adopting a service-oriented view for modeling and software engineering. Secondly, a service-oriented implementation of the provenance infrastructure simplifies its integration into a SOA/OGSA, thus promoting the adoption of the infrastructure in service-based applications. Finally, a service-oriented provenance infrastructure deploys easily into heterogeneous distributed environments, thus facilitating the access, sharing and reuse of provenance data.

The hybrid approach to provenance data collection, storage and query are flexible and pragmatic. Semantic metadata contain rich semantic and knowledgeable information by which users can perform reasoning or mining to derive added values or discover implicit knowledge. In contrary, execution data are usually raw data, containing little semantic information. Practically the hybrid approach is easy to be implemented by marrying the state of the art of the Semantic Web and database management technologies.

The benefits of developing a reference augmented provenance system in GEODISE are multiple. Firstly, it helps pin down the conception, modeling and representation of augmented provenance. Secondly, it helps capture user requirements for and characteristics of provenance in the context of service-based applications. Thirdly, it helps identify software requirements for a provenance system, i.e. what a provenance system has to do. Fourthly, the successful design, implementation and operation of the provenance system, though still preliminary, have demonstrated our conception of provenance, its design approaches and implementation rationale. Finally, it helps identify a number of problems and motivate the discovery of possible solutions.

We also learn lessons from the deployment: First, tools should be provided for end users in their familiar working environments. Second, easy-to-use tools should hide as much technical details as possible that are not relevant to the end users.

VI. CONCLUSIONS

In this paper we have analysed the nature of service-oriented computing and elicited the conception of augmented provenance from a real world application scenario. We have proposed a SWS-based hybrid approach for managing augmented provenance based on the latest technologies in the Semantic Web, ontologies, and SWS. We have described a system architecture that specifies the core components and functionalities for managing the lifecycle of augmented provenance. The proposed approach and architecture have been implemented in the context of GEODISE project, which produced a suite of generic APIs and front-end GUIs that are

applicable for the realisation of provenance systems for other application domains.

Although our work is still in its early stage, the conception of augmented provenance and SWS-based approach are innovative and inspiring: provenance will be an indispensable ingredient in the future Web; and reusing SWS's semantic descriptions for provenance is a good example of the Semantic Web applications. By the GEODISE example we have shown how provenance system can be designed and used for problem solving. Further investigation will focus on the granularity of provenance data, and its use to support trust and security.

ACKNOWLEDGMENT

This work is based on the UK EPSRC GEODISE e-Science pilot project (GR/R67705/01) and EU FP6 PROVENANCE project. The authors gratefully acknowledge the contributions of Dr. William Cheung for his insightful and inspiring comments and suggestions.

REFERENCES

[1] Foster, I., Kesselman, C., Nick, J., Tuecke, S. (2002), Grid Services for Distributed System Integration, Computer, 35(6), 37-46

[2] Cui, Y., Widom, J. and Wiener, J.L. (2000), Tracing the Lineage of View Data in a Warehousing Environment. ACM Trans. on Database Systems, 25(2):179–227

[3] Buneman, P., Khanna, S. and Tan, W.C. (2001), Why and Where: A Characterization of Data Provenance. In Proceedings of 8th International Conference on Database Theory, pp316–330

[4] Foster, I., Vockler, J., Wilde, M., Zhao, Y. (2002), Chimera: A virtual data system for representing, querying, and automating data derivation, Proceedings of the 14th International Conference on Scientific and Statistical Database Management, pp37-46

[5] Boss, R. (2002). A conceptual framework for composing and managing scientific data lineage, Proceedings of the 14th International Conference on Scientific and Statistical Database Management, pp47-55

[6] Buneman P., Chapman A. and Cheney J. (2006), Provenance management in curated databases. SIGMOD Conference 2006: 539-550

[7] da Silva, P.P., McGuinness, D.L. and McCool, R. (2003), Knowledge Provenance Infrastructure. IEEE Data Engineering Bulletin Vol.26 No.4, pp26-32

[8] McGuinness D.L. and da Silva P.P. (2004), Explaining Answers from the Semantic Web: The Inference Web Approach, Journal of Web Semantics, Vol.1, No.4, pp1-27

[9] Workshop on Data Derivation and Provenance, (2002), http://www-p.mcs.anl.gov/~foster/provenance/

[10] Workshop on Data Provenance and Annotation, (2003), http://www.nesc.ac.uk/esi/events/304/

[11] International Provenance and Annotation Workshop IPAW'06, (2006), http://www.ipaw.info/ipaw06/

[12] Zhao, J., Wroe, C., Goble, C., Stevens, R., Quan D., and Greenwood M. (2004). Using Semantic Web Technologies for Representing e-Science Provenance, LNCS, No.3298, pp92-106

[13] Moreau, L., Chen, L., Groth, P., Ibbotson, J., Luck, M., Miles, M., Rana, O., Tan, V., Willmott, S. and Xu, F. (2005). Logical architecture strawman for provenance systems, Technical report, University of Southampton.

[14] Jiao, Z., Wason, J.L., Song, W., Xu, F., Eres, H., Keane, A.J., and Cox, S.J. (2004). Databases, Workflows and the Grid in a Service Oriented Environment, Euro-Par2004, Parallel Processing, LNCS, No.3149, pp972-979.

[15] Harris, S., Gibbins, N. (2003). 3store: Efficient Bulk RDF Storage. Proceedings of 1st International Workshop on Practical and Scalable Semantic Systems, pp1-15.

[16] Horrocks, I., Li, L., Turi, D., Bechhofer, S. (2004). The instance store: DL reasoning with large numbers of individuals, Proceedings of the 2004 Description Logic Workshop, pp31-40

[17] Chen, L., S. J. Cox, C. Goble, A. J. Keane, A. Roberts, N. R. Shadbolt, P. Smart, and F. Tao (2002). Engineering knowledge for engineering grid applications. In Proceedings of Euroweb 2002 Conference, The Web and the GRID: From e-science to e-business, pp12–25.

[18] Haarslev, V., Möller, R. (2003). Racer: A Core Inference Engine for the Semantic Web, Proceedings of the 2nd International Workshop on Evaluation of Ontology-based Tools (EON2003), pp27-36.

[19] Chen, L., Shadbolt N.R., Tao F. and Goble C. (2006). Managing Semantic Metadata for Web/Grid Services, International Journal of Web Service Research, in press.

Genetic Programming for Object Detection: Improving Fitness Functions and Optimising Training Data

Mengjie Zhang, *Member, IEEE,* Malcolm Lett

***Abstract*— This paper describes an approach to the improvement of a fitness function and the optimisation of training data in genetic programming (GP) for object detection particularly object localisation problems. The fitness function uses the weighted F-measure of a genetic program and considers the localisation fitness values of the detected object locations. To investigate the training data with this fitness function, we categorise the training data into four types: *exact centre, close to centre, include centre,* and *background.* The approach is examined and compared with an existing fitness function on three object detection problems of increasing difficulty. The results suggest that the new fitness function outperforms the old one by producing far fewer false alarms and spending much less training time and that the first two types of the training examples contain most of the useful information for object detection. The results also suggest that the complete background type of data can be removed from the training set.**

***Index Terms*— Genetic programming, object detection, object localisation, object recognition, object classification, evolutionary computing, fitness function, training data.**

I. Introduction

OBJECT detection tasks arise in a very wide range of applications, such as detecting faces from video images, finding tumours in a database of x-ray images, and detecting cyclones in a database of satellite images [1], [2], [3], [4]. In many cases, people (possibly highly trained experts) are able to perform the detection task well, but there is either a shortage of such experts, or the cost of people is too high. Given the amount of image data containing objects of interest that need to be detected, computer based object detection systems are of immense social and economic value.

An object detection program must automatically and correctly determine whether an input vector describing a portion of a large image at a particular location in the large image contains an object of interest or not and what class the suspected object belongs to. Writing such programs is usually difficult and often infeasible: human programmers often cannot identify all the subtle conditions needed to distinguish between all objects and background instances of different classes.

Genetic programming (GP) is a relatively recent and fast developing approach to automatic programming [5], [6], [7]. In GP, solutions to a problem can be represented in different forms but are usually interpreted as computer programs. Darwinian principles of natural selection and recombination are used to evolve a population of programs towards an effective solution to specific problems. The flexibility and expressiveness of computer program representation, combined with the powerful capabilities of evolutionary search, make GP an exciting new method to solve a great variety of problems. GP has been applied to a range of object detection and recognition tasks with some success [5], [8], [9], [10], [11], [12], [13].

Finding a good fitness function for a particular object detection problem is an important but difficult task in developing a GP system. Various fitness functions have been devised for object detection, with varying success [5], [9], [11], [14], [15]. These tend to combine many parameters using scaling factors which specify the relative importance of each parameter, with no obvious indication of what scaling factors are good for a given problem. Many of these fitness functions require clustering to be performed to group multiple localisations of single objects into a single point before the fitness is determined [16], [15], [14]. Other measures are then incorporated in order to include information about the pre-clustered results (such as how many points have been found for each object). While some of these systems achieved good detection rates, many of them resulted in a large number of false alarms. In addition, the clustering process during the evolutionary process made the training time very long.

Organising training data is critical to any learning approaches. The previous approaches in object detection tend to use all possible positions of the large image in training an object detector. However, this usually requires a very long training time due to the use of a large number of positions on the background.

This paper aims to investigate a new fitness function and a new way to optimise the training data in GP for object detection, in particular object localisation, with the goal of improving the detection performance and refining training examples. The approach will be examined and compared with an existing GP approach on a sequence of object detection problems of increasing difficulty.

The remainder of this paper is organised as follows. Section II gives some essential background on GP and object detection/recognition. Section III describes the GP approach to object detection, including the major components of the approach. Section IV focuses on the new fitness function and

Mengjie Zhang is with the School of Mathematics, Statistics and Computer Science, Victoria University of Wellington, P. O. Box 600, Wellington, New Zealand (phone: +64 4 463 5654; fax: +64 4 463 5045; email: mengjie@mcs.vuw.ac.nz).

Malcolm Lett is also with the School of Mathematics, Statistics and Computer Science, Victoria University of Wellington, P. O. Box 600, Wellington, New Zealand.

compares it with an existing clustering based fitness function. Section V investigates the training data. Finally, we draw conclusions in section VI. Some GP basics are given in the appendix.

II. Background

A. Genetic Programming and Main Characteristics

GP is an approach to automatic programming, in which a computer can construct and refine its own programs to solve specific tasks. First introduced by Koza [6] in the early 1990s, GP has become another main genetic paradigm in evolutionary computation (EC) in addition to the well known *genetic algorithms* (GAs).

Compared with GAs, GP has a number of characteristics. While the standard GAs use bit strings to represent solutions, the forms evolved by GP are generally trees or tree-like structures. The standard GA bit strings use a fixed length representation while the GP trees can vary in length. While the GAs use a binary alphabet to form the bit strings, the GP uses alphabets of various sizes and content depending on the problem domain. These trees are made up of internal nodes and leaf nodes, which have been drawn from a set of primitive elements that are relevant to the problem domain. Compared with a bit string to represent a given problem, the trees can be much more flexible.

The basic concepts, genetic operators, and the GP algorithm are described in the appendix.

B. Object Detection

The term *object detection* here refers to the detection of small objects in large images. This includes both *object classification* and *object localisation*. *Object classification* refers to the task of discriminating between images of different kinds of objects, where each image contains only one of the objects of interest. *Object localisation* refers to the task of identifying the positions of all objects of interest in a large image. The object detection problem is similar to the commonly used terms *automatic target recognition* and *automatic object recognition*.

Object detection performance is usually measured by *detection rate* and *false alarm rate*. The detection rate (DR) refers to the number of small objects correctly reported by a detection system as a percentage of the total number of actual objects in the image(s). The false alarm rate (FAR), also called false alarms per object [17], refers to the number of non-objects incorrectly reported as objects by a detection system as a percentage of the total number of actual objects in the image(s). Note that the detection rate is between 0 and 100%, while the false alarm rate may be greater than 100% for difficult object detection problems.

C. GP Related Work for Object Detection and Recognition

Since the early 1990s, there has been only a small amount of work on applying GP techniques to object classification, object detection and other image recognition problems. This in part reflects the fact that GP is a relatively young discipline compared with, say, neural networks and genetic algorithms.

In terms of the number of classes in object detection, there are two categories. The first is *one-class object detection problem*, where there are multiple objects in each image, however they belong to or are considered the same (single) class of interest. In nature, these problems contain a two-class (binary) classification problem: *object* versus *non-object*, also called *object* versus *background*. Examples are detecting small targets in thermal infrared images [17] and detecting a particular face in photograph images [18]. The problem is actually the same as *object localisation*, where the main goal is to find where the objects of interest are in the large images. The second is *multi-class object detection problem*, where there are multiple object classes of interest each of which has multiple objects in each image. Detection of handwritten digits in postal code images [19] is an example of this kind. While GP has been widely applied to the one-class object detection and binary classification problems [15], [8], [9], [20], it has also been applied to multi-class object detection and classification problems [21], [22], [23], [10], [24], [11].

In terms of the representation of genetic programs, different forms of genetic programs have been developed in GP systems for object classification and image recognition. The main program representation forms include tree or tree-like or numeric expression programs [5], [7], [21], [11], graph based programs [5], linear GP [25], linear-graph GP [26], and grammar based GP [27].

The use of GP in object detection and image recognition has also been investigated in a variety of application domains. These domains include military applications [9], [20], English letter recognition [28], face/eye detection and recognition [29], [22], [30], vehicle detection [15], [31] and other vision and image processing problems [32], [33], [6], [34], [35], [36].

III. The GP Approach to Object Detection

The process for object detection is shown in Figure 1. A raw image is taken and a trained localiser applied to it, producing a set of points found to be the positions of these objects. Single objects could have multiple positions ("localisations"), however ideally there would be exactly one localisation per object. Regions of the image are then "cut out" at each of the positions specified. Each of these cutouts are then classified using the trained classifier.

This method treats all objects of multiple classes as a single "object of interest" class for the purpose of localisation, and the classification stage handles attaching correct class labels. Compared with the single-stage approach [10], [11], this approach has the advantage that the training is easier for both stages as a specific goal is focused on the training of each of the two stages. The first is tailored to achieving results as close to the object centres as possible (to achieve high "positional accuracy"), while the second is tailored to making all classifications correct (high "classification accuracy").

The object localisation stage is performed by means of a window which sweeps over the whole image, and for each position extracts the features and passes them to the trained localiser. The localiser then determines whether each position is an object or not (i.e. background).

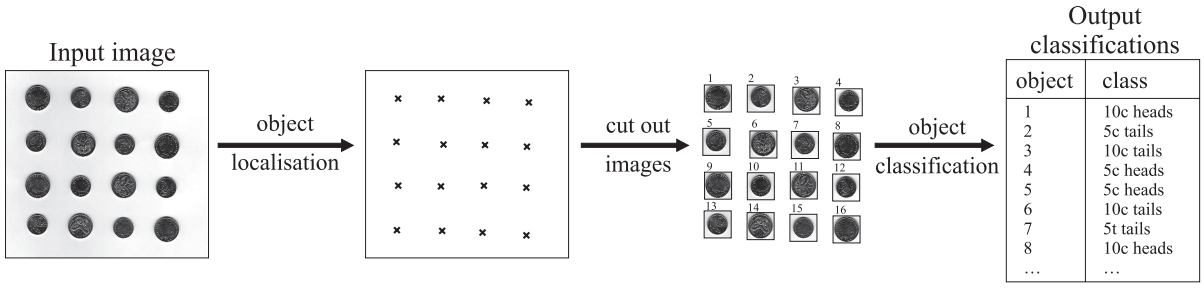


Fig. 1. An overview of the object detection process.

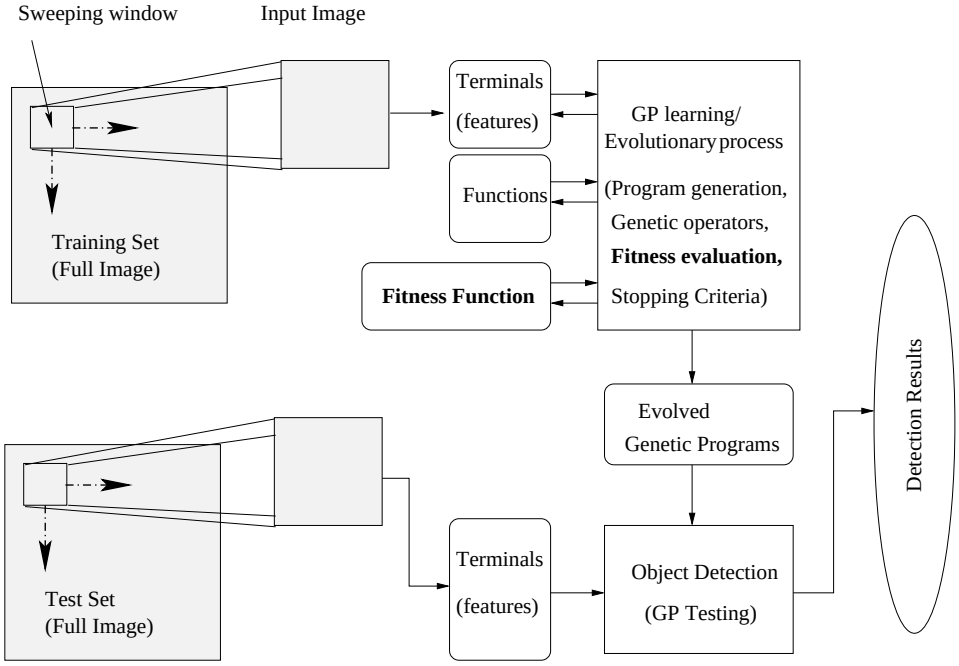


Fig. 2. GP approach to object detection.

Our work will focus on object localisation using genetic programming. Figure 2 shows an overview of this approach, which has a learning process and a testing procedure. In the learning/evolutionary process, the evolved genetic programs use a square input field which is large enough to contain each of the objects of interest. The programs are applied at many sampled positions within the images in the *training set* to detect the objects of interest. If the program localiser returns a value greater than or equal to zero, then this position is considered the centre of an object of interest; otherwise it is considered background. In the test procedure, the best evolved genetic program obtained in the learning process is then applied, in a moving window fashion, to the whole images in the *test set* to measure object detection performance.

This approach has five major components: (1) Determination of a terminal set; (2) Determination of a function set; (3) Construction of a new fitness function; (4) Determination of the major parameter values and the termination criteria; and (5) investigation of the training data. In addition, to examine the performance of this approach, we also need to choose the object detection example tasks.

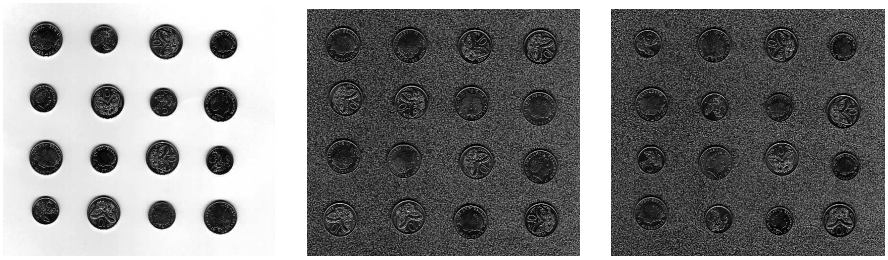
Construction of a new fitness function and investigation of the training data are the main focuses of this paper, which will be described in the next sections. In the rest of this section, we will describes all of the other components.

A. Terminal Set

For object detection problems, terminals generally correspond to image features. In this approach, the features are extracted by calculating the mean and standard deviation of pixel values within several circular regions. This set of features has the advantages of being rotationally invariance. In addition, we also used a constant terminal. Note that finding a good set of features is beyond the goal of this paper, and we will use this set of features to check the performance of both the existing and the new approaches for comparison purpose only.

B. Function Set

The function set contains the four standard arithmetic and a conditional operation: $FuncSet = \{+, -, *, /, if\}$. The $+$, $-$, and $*$ operators are usual addition, subtraction and multiplication, while / represents "protected" division. The *if* function returns its second argument if the first argument is positive or returns its third argument otherwise.



(a) (b) (c)

Fig. 3. Sample images in the three data sets. (a) Easy; (b) Medium difficulty; (c) Hard.

C. GP Structure, Parameters and Termination Criteria

In this system, we used tree structures to represent genetic programs [6]. The ramped half-and-half method [5] was used for generating programs in the initial population and for the mutation operator. The proportional selection mechanism and the reproduction, crossover and mutation operators were used in evolution.

We used a population of 500 genetic programs for evolution in each experiment run. The reproduction rate, crossover rate and mutation rate were 5%, 70% and 25%, respectively. The program size was initialised to 4 and it could increase to 8 during evolution.

The system run 50 generations unless it successfully found an ideal solution or the performance on the validation set fell down, in which cases the evolution was terminated early.

D. Data Sets

To investigate the performance of this approach, we chose three image data sets of New Zealand 5 and 10 cent coins in the experiments. Examples are shown in Figure 3. The data sets are intended to provide object localisation/detection problems of increasing difficulty. The first data set (*easy*) contains images of tails and heads of 5 and 10 cent coins against an almost uniform background. The second (*medium difficulty*) is of 10 cent coins against a noisy background, making the task harder. The third data set (*hard*) contains tails and heads of both 5 and 10 cent coins against a noisy background.

We used 24 images for each data set in our experiments and equally split them into three sets: a training set for learning good genetic programs, a validation set for monitoring the training process to avoid overfitting, and a test set to measure object detection performance.

In our experiments, a total number of 100 runs were performed on each data set and the average results are presented in the next two sections.

IV. Fitness Function

A. Design Considerations

During the evolutionary process for object detection, we expect that the evolved genetic programs only detect the objects when the sweeping window is centred over these objects. However, in the usual case, these evolved genetic programs will also detect some "objects" not only when the sweeping window is within a few pixels of the centre of the target objects, but also when the sweeping window is centred over a number of cluttered pieces of background. Clearly, these "objects" are not those we expected but false alarms.

Different evolved genetic programs typically result in different numbers of false alarms and such differences should be reflected when these programs are evaluated by the fitness function.

When designing a fitness function for object detection problems, a number of considerations need to be taken into account. At least the following requirements should be considered.

- R1. The fitness function should encourage a greater number of objects to be detected. In the ideal case, all the objects of interest in large images can be detected.
- R2. The fitness function should prefer a fewer number of false alarms on the background.
- R3. The fitness function should encourage genetic programs to produce detected object positions closer to the centres of the target objects.
- R4. For a single object to be detected, the fitness function should encourage programs to produce fewer detected "objects" (positions) within a few pixels from the target centre.
- R5. For two programs which produce the same number of detected "objects" for a single target object but the "objects" detected by the first program are closer to the target object centre than those detected by the second program, the fitness function should rank the first program better than the second.

Some typical examples of these requirements are shown in figure 4. In this figure, the circles are target objects and squares are large images or regions. A cross (x) represents a detected object. In each of the five cases, the program associated with the left figure should be considered better than that with the right.

B. An Existing Fitness Function

As the goal is to detect the target objects with no or a small number of false alarms, many GP systems uses a combination of detection rate and false alarm rate or recall and precision as the fitness function. For example, a previous GP system uses the following fitness function [10]:

$$fitness_{CBF} = A \cdot (1 - DR) + B \cdot FAR + C \cdot FAA \quad (1)$$

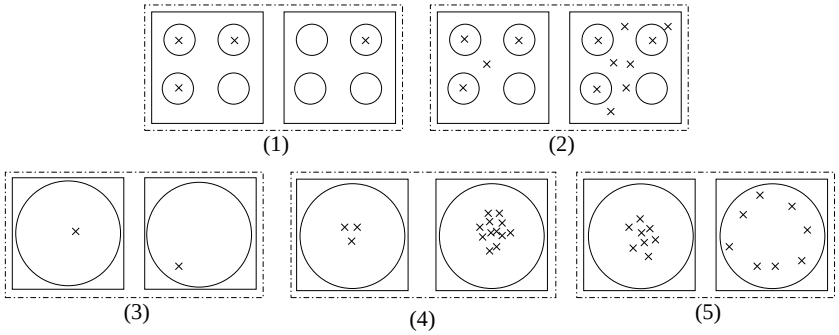


Fig. 4. Examples of the design considerations of the fitness function.

where DR, FAR, and FAA are detection rate (the number of small objects correctly reported by a detection system as a percentage of the total number of actual objects in the images), false alarm rate (also called *false alarms per object*, the number of non-objects incorrectly reported as objects by a detection system as a percentage of the total number of actual objects in the images), and false alarm area (the number of false alarm pixels which are not object centres but are incorrectly reported as object centres before clustering), respectively, and A, B, C are constant weights which reflect the relative importance of detection rate versus false alarm rate versus false alarm area.

Basically, this fitness function has considered requirement 1, and partially considered requirements 2 and 4, but does not take into accounts of requirements 3 and 5. Although this fitness function performed reasonably well on some problems, it still produced many false alarms and the evolutionary training time was still very long [10]. Since this method used clustering before calculating the fitness, we refer to it as *clustering based fitness*, or CBF for short.

C. A New Fitness Function — RLWF

To avoid a very large false alarm rate (greater than 100% for difficult problems) in the training process, we use precision and recall, both of which have the range between [0, 1], to construct the new fitness functions. *Precision* refers to the number of objects correctly localised/detected by a GP system as a percentage of the total number of object localised/detected by the system. *Recall* refers to the number of objects correctly localised by a system as a percentage of total number of target objects in a data set. Note that precision/recall and detection rate/false alarm rate have internal relationship, where the value of one pair for a problem can be calculated using the other for the same problem.

During the object localisation process, a genetic program might consider many pixel positions in an image as object centres and we call each object centre localised in an image by a genetic program a *localisation*.

Unlike the previous fitness function CBF, the new fitness function is based on a "Relative Localisation Weighted F-measure" (RLWF), which attempts to acknowledge the worth/goodness of individual localisations made by the genetic program. Instead of using either correct or incorrect to represent a localisation, each localisation is allocated a weight (referred to as the *localisation fitness*, *LF*) which represents its individual worth and counts towards the overall fitness.

Each weight is calculated based on its relative location, or the distance of the localisation from the centre of the closest object, as shown in Equation 2.

$$\mathrm{LF}(x,y) = \begin{cases} 1 - \frac{\sqrt{x^2+y^2}}{r} & , \text{ if } \sqrt{x^2+y^2} \le r \\ 0 & , \text{ otherwise} \end{cases} \quad (2)$$

where $\sqrt{x^2+y^2}$ is the distance of the localisation position (x, y) from target object centre, and *r* is called the "localisation fitness radius", defined by the user. In this system, r is set to a half of the square size of the input window, which is also the radius of the largest object.

In order to deal with all the situations in the five design requirements, we used the localisation fitness to construct our new fitness function, as shown in Equations 3 to 5. The precision and recall are calculated by taking the localisation fitness for all the localisations of each object and dividing this by the total number of localisations or total number of target objects respectively.

$$\mathrm{WP} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{L_i}\mathrm{LF}(x_{ij}, y_{ij})}{\sum_{i=1}^{N} L_i} \quad (3)$$

$$\mathrm{WR} = \frac{\sum_{i=1}^{N}\frac{\sum_{j=1}^{L_i}\mathrm{LF}(x_{ij}, y_{ij})}{L_i}}{N} \quad (4)$$

$$\mathrm{fitness}_{RLWF} = \frac{2 \times \mathrm{WP} \times \mathrm{WR}}{\mathrm{WP} + \mathrm{WR}} \quad (5)$$

where N is the total number of target objects, (x_{ij}, y_{ij}) is the position of the j-th localisation of object i, L_i is number of localisations made to object i, WP and WR are the weighted precision and recall, and $\mathit{fitness}_{RLWF}$ is the localisation fitness weighted F-measure, which is used as the new fitness function.

The new fitness function has a number of properties. Firstly, the main parameter in this fitness function is the *localisation fitness*, which can by easily determined in the way presented here. This has an advantage over the existing methods which have many parameters whose values usually need to be manually determined. Secondly, in the previous approaches, the multiple localisations of each object must be clustered into

TABLE I
RESULTS OF THE GP SYSTEMS WITH THE TWO FITNESS FUNCTIONS.

Dataset	Fitness function	Test Accuracy			Training Efficiency	
		LR (%)	LP (%)	ExtraLocs	Generations	time(sec)
Easy	CBF	99.99	98.26	324.09	13.69	178.99
	RLWF	99.99	99.36	98.35	36.44	111.33
Medium	CBF	99.60	83.19	804.88	36.90	431.94
	RLWF	99.90	94.42	95.69	34.35	105.56
Hard	CBF	98.22	75.54	1484.51	31.02	493.65
	RLWF	99.53	87.65	114.86	33.27	107.18

one group and its centre found. While this is not a too difficult task, it is very time consuming to do during training. This new fitness function does not require clustering before the fitness is calculated. We expect that the new fitness function can do a better job in terms of reducing false alarms and evolutionary training time.

D. Results

To give a fair comparison for the two fitness functions, the "localisation recall (LR) and precision (LP)" were used to measure the final object detection accuracy on the test set. LR is the number of objects with one or more correct localisations within the localisation fitness radius at the target object centres as a percentage of the total number of target objects, and LP is the number of correct localisations which fall within the localisation radius at the target object centres as a percentage of the total number of localisations made. In addition, we also check the "Extra Localisations" (ExtraLocs) for each system to measure how many extra localisations were made for each object. The training efficiency of the systems is measured with the number of training generations and the CPU (user) time in second.

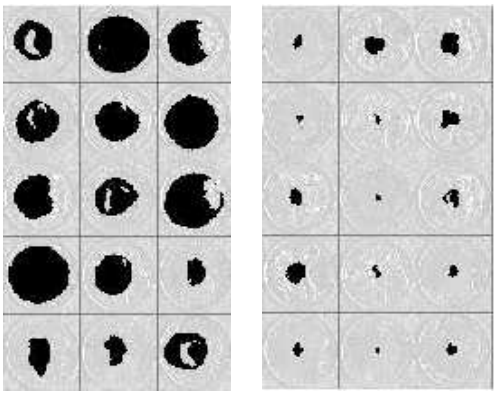
Table I shows the results of the GP systems with the two fitness functions. The results on the easy data set show that both the fitness functions achieved almost perfect test accuracy. Almost all the objects of interest in this data set were successfully localised with very few false alarms (both LR and LP are very close to 100%), reflecting the fact that the detection task in this data set is relatively easy. However, the extra locations and the training time resulted from the two approaches are quite different. The new fitness function (RLWF) produced a far fewer number of extra localisations per object than clustering based fitness function (CBF) and the gap between them is significant. Although the CBF approach used only 13.69 generations on average, which are considerably fewer than that of the new RLWF, it actually spent about 50% longer training time. This confirms our early hypothesis that the clustering process in the CBF approach is time consuming and the approach with the new fitness function is more efficient than that with CBF.

The results on the other two data sets show a similar pattern in terms of the number of extra localisations and training time. The systems with RLWF always produced a significantly fewer number of extra localisations and a much short training time than CBF. In addition, although almost all the objects of interest in the large images were successfully detected (LRs are almost 100%), the localisation precisions achieved by RLWF were significantly better than CBF, suggesting that the new fitness function outperforms the existing one in terms of reducing false alarms.

As expected, performance on the three data sets deteriorated as the degree of difficulty of the object detection problem was increased.

E. Detection Map Analysis

To give an intuitive view of detection performance of the two fitness functions, we checked the "detection maps" of some objects in the test set. Figures 5 (a) and (b) show the detection maps for the same 15 objects in the medium difficulty data set produced by the two approaches. The black pixels in these maps indicate the localisations of the 15 objects produced using the two fitness functions. The "background" means that no objects were found in those positions.



(a) (b)

Fig. 5. Sample object detection maps. (a) CBF; (b) RLWF.

As shown in the figure, the clustering based fitness function CBF resulted in a huge number of extra localisations for all the 15 objects detected. The new fitness function, however, only resulted in a small number of extra localisations. These maps confirm that the new fitness function was more effective than the clustering based fitness function on these problems.

V. OPTIMISING TRAINING DATA

We could train the detection system with a full set of cutouts taken from a window at all possible positions over the training images. However, for a set of large images, this can create a huge number of training examples making the training time unsuitably long. While we can reduce the total number of training examples using a combination of hand-chosen and

randomly chosen examples [16], in this approach, we focus on investigating whether some examples are better than others and how we pick up better examples.

A. Four Training Data Types

The traditional approaches usually use *positive* and *negative* examples. The former refers to the exact object examples and the latter refers to those for the background [9], [10], [15]. However, this did not consider those with a portion of objects and a portion of background. In this approach, we identified four basic types of training examples, as shown in figure 6. The *exact centre* type (figure 6a) refers to the positive object examples which sit exactly the centre of the sweeping window. This type of examples has only a very small number. For example, in each of our training images, we have only 16 such examples out of approximately half a million pixel positions. The *background* type (figure 6d) refers to the positions (x) which do not contain any piece of objects. This type typically has a huge number of examples. The *close to centre* type refers to the examples that have the centre of the sweeping window falling down within the bounds of an object (figure 6b). The *include objects* type refers to the examples that contain some pixels of an object but are not considered as the *close to centre* type.

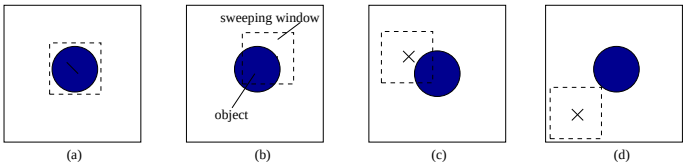


Fig. 6. Examples of training data types caused by different input window positions.

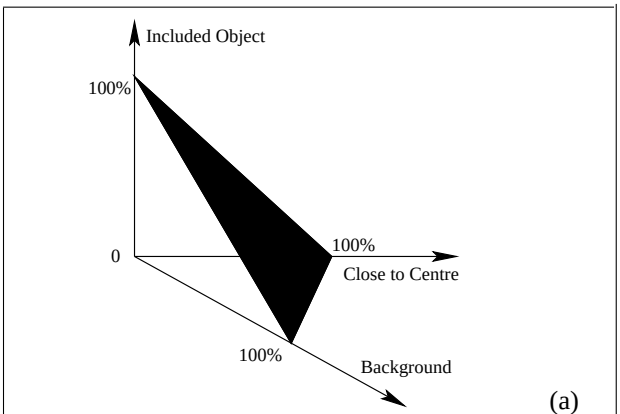
B. Optimisation of Training Data

For a problem domain, we assume that there is some proportion of these four types which is optimal (or close to optimal) for object detection. From previous research, we found that the exact centre type is always important for object detection. As the number of examples of this type is very small, we will always use this type of examples in the experiments and assume that the best results can only be achieved by including them. In the remainder of investigation, we will vary the proportions among the rest three types to find the optimal combinations.

Based on this idea, if we use C, I and B to refer to percentage of the examples for the three types *close to centre*, *include objects* and *background*, then we have:

$$C + I + B = 100\%$$

This has the nice feature that it represents only a plane effectively reducing the parameter search space from 3D to 2D, as shown in figure 7 (a). We experimented with 28 separate proportions sampled from the plane in figure 7 (a), as shown in figure 7 (b), where each entry represents value for I for a given C and B. For example, the first two entries in the first row show that, using no background ($B = 0$), we will examine 100% C with 0% I, and 83% C with 17% I type objects, respectively.



B\C	I for each C and B (%)						
	100	83	67	50	33	17	0
0	0	17	33	50	67	83	100
17		0	17	33	50	67	83
33			0	17	33	50	67
50				0	17	33	50
67					0	17	33
83						0	17
100							0

(b)

Fig. 7. Training data proportions set.

C. Results

For each experiment with a sampled proportion, we did 100 runs. These were made up of 10 different random seeds when extracting the training data from source images, by 10 different random seeds for the GP system. Other parameters are the same as before.

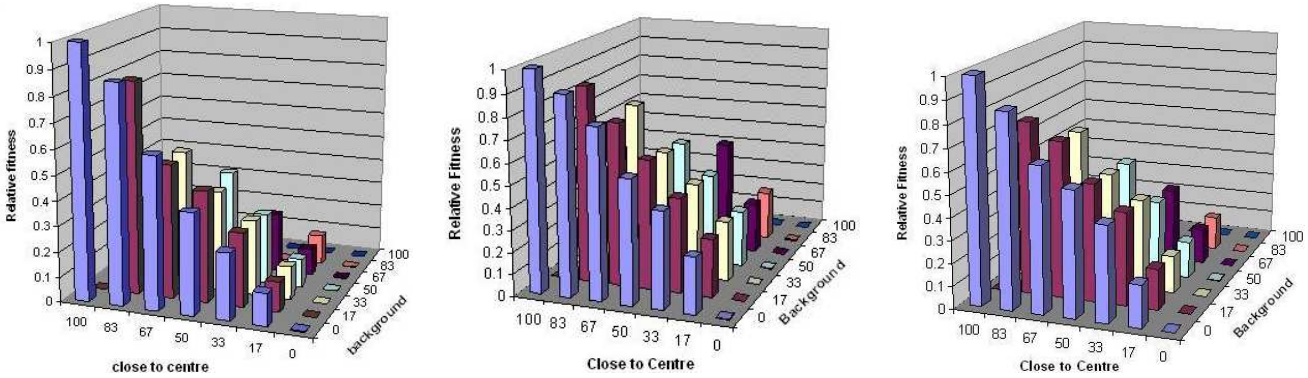
The average results on the *test set* are shown in figure 8. In the figure, the x and y axes are the C and B, and the z is the *relative* fitness for the these problems (1.0 or 100% means the ideal case).

As shown in the figure, for all the three data sets, the value of C, or the percentage of the objects for the *Close to Centre* type played an important role using our new fitness function. The best detection results were achieved with 100% examples for the *close to centre* type and the worst results were produced when we do not use any example in this type at all. The more object examples used in this type, the best results achieved. However, the *Background* type objects were not critical for these data sets. These examples did not seem to have clear bad or good influence.

These results suggest that, when using the new RLWF fitness function for these object detection, good fitness results can be achieved with only the two types, *Exact Centre* and *Close to Centre*, and most if not all object examples for the other two types *Include Object* and *Background* can be taken out from the training set.

Inspection of this reveals that, this is not only because the first two types of objects might contain the most useful information for object detection, but more importantly, because the new RLWF fitness function is capable of learning well from these two types of examples and can cope well with the goal of finding object centres from large images. This is mainly due to the fact that the RLWF fitness function consider the relative effect of the detected "objects" in different locations.

A further inspection of the use of the old fitness function



(a) (b) (c)

Fig. 8. Results of optimisation. (a) Easy; (b) Medium difficulty; (c) Hard.

reveals that the old fitness function must use object examples from all the four types. This is because the old fitness function cannot capture the relative effect information from the objects of the first two types only. This also suggests that the new fitness function is more effective than the old one for object detection, particularly when only are training examples from the first two types available.

VI. Conclusions

The goal of this paper was to develop a new fitness function for object detection and investigate its influence on optimising the training data. Rather than using a clustering process to determine the number of objects detected by the GP systems, the new fitness function introduced a weight called localisation fitness to represent the goodness of the detected objects and used weighted F-measures. To investigate the training data with this fitness function, we categorise the training data into four types. This approach is examined and compared to that with the old clustering based fitness function on three coin detection problems of increasing difficulty.

The results suggest that the new fitness function outperforms the old one by producing far fewer false alarms and spending much less training time. Further investigation on the four types of the training object examples suggests that the first two types of objects can be used to produce good detection results and that the new fitness function is effective in optimising the training data for object detection.

In the future, we will apply the new approach to other object detection problems particularly with non-circular objects.

Acknowledgement

This work was supported in part by the Marsden Fund at Royal Society of New Zealand under grant No. 05-VUW-017 and University Research Fund 6/9 at Victoria University of Wellington.

References

[1] P. D. Gader, J. R. Miramonti, Y. Won, and P. Coffield, "Segmentation free shared weight neural networks for automatic vehicle detection," *Neural Networks*, vol. 8, no. 9, pp. 1457–1473, 1995.

[2] H. L. Roitblat, W. W. L. Au, P. E. Nachtigall, R. Shizumura, and G. Moons, "Sonar recognition of targets embedded in sediment," *Neural Networks*, vol. 8, no. 7/8, pp. 1263–1273, 1995.

[3] M. W. Roth, "Survey of neural network technology for automatic target recognition," *IEEE Transactions on neural networks*, vol. 1, no. 1, pp. 28–43, March 1990.

[4] A. M. Waxman, M. C. Seibert, A. Gove, D. A. Fay, A. M. Bernandon, C. Lazott, W. R. Steele, and R. K. Cunningham, "Neural processing of targets in visible, multispectral ir and sar imagery," *Neural Networks*, vol. 8, no. 7/8, pp. 1029–1051, 1995.

[5] W. Banzhaf, P. Nordin, R. E. Keller, and F. D. Francone, *Genetic Programming: An Introduction on the Automatic Evolution of computer programs and its Applications*. San Francisco, Calif. : Morgan Kaufmann Publishers; Heidelburg : Dpunkt-verlag, 1998, subject: Genetic programming (Computer science); ISBN: 1-55860-510-X.

[6] J. R. Koza, *Genetic programming : on the programming of computers by means of natural selection*. London, England: Cambridge, Mass. : MIT Press, 1992.

[7] ——, *Genetic Programming II: Automatic Discovery of Reusable Programs*. London, England: Cambridge, Mass. : MIT Press, 1994.

[8] A. Song, V. Ciesielski, and H. Williams, "Texture classifiers generated by genetic programming," in *Proceedings of the 2002 Congress on Evolutionary Computation CEC2002*, D. B. Fogel, M. A. El-Sharkawi, X. Yao, G. Greenwood, H. Iba, P. Marrow, and M. Shackleton, Eds. IEEE Press, 2002, pp. 243–248.

[9] W. A. Tackett, "Genetic programming for feature discovery and image discrimination," in *Proceedings of the 5th International Conference on Genetic Algorithms, ICGA-93*, S. Forrest, Ed. University of Illinois at Urbana-Champaign: Morgan Kaufmann, 17-21 July 1993, pp. 303–309.

[10] M. Zhang, P. Andreae, and M. Pritchard, "Pixel statistics and false alarm area in genetic programming for object detection," in *Applications of Evolutionary Computing, Lecture Notes in Computer Science, LNCS Vol. 2611*, S. Cagnoni, Ed. Springer-Verlag, 2003, pp. 455–466.

[11] M. Zhang, V. Ciesielski, and P. Andreae, "A domain independent window-approach to multiclass object detection using genetic programming," *EURASIP Journal on Signal Processing, Special Issue on Genetic and Evolutionary Computation for Signal Processing and Image Analysis*, vol. 2003, no. 8, pp. 841–859, 2003.

[12] M. Zhang, X. Gao, and W. Lou, "Looseness controlled crossover in gp for object classification," in *Proceedings of IEEE Congress on Evolutionary Computation, a part of IEEE Congress on Computational Intelligence*, S. Lucas, Ed., Vancouver BC, Canada, 16–21 July 2006, pp. 4428–4435.

[13] M. Zhang and W. Smart, "Using gaussian distribution to construct fitness functions in genetic programming for multiclass object classification," *Pattern Recognition Letters*, vol. 27, no. 11, pp. 1266–1274, Aug. 2006, evolutionary Computer Vision and Image Understanding.

[14] W. Smart and M. Zhang, "Classification strategies for image classification in genetic programming," in *Proceeding of Image and Vision Computing Conference*, D. Bailey, Ed., Palmerston North, New Zealand, November 2003, pp. 402–407.

[15] D. Howard, S. C. Roberts, and R. Brankin, "Target detection in SAR

imagery by genetic programming," *Advances in Engineering Software*, vol. 30, pp. 303–311, 1999.
[16] U. Bhowan, "A domain independent approach to multi-class object detection using genetic programming," BSc Honours research thesis, School of Mathematical and Computing Sciences, Victoria University of Wellington, 2003.
[17] M. V. Shirvaikar and M. M. Trivedi, "A network filter to detect small targets in high clutter backgrounds," *IEEE Transactions on Neural Networks*, vol. 6, no. 1, pp. 252–257, Jan 1995.
[18] S.-H. Lin, S.-Y. Kung, and L.-J. lin, "Face recognition/detection by probabilistic decision-based neural network," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 114–132, Jan 1997.
[19] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. H. W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, pp. 541–551, 1989.
[20] W. A. Tackett, "Recombination, selection, and the genetic construction of computer programs," Ph.D. dissertation, Faculty of the Graduate School, University of Southern California, Canoga Park, California, USA, April 1994.
[21] T. Loveard and V. Ciesielski, "Representing classification problems in genetic programming," in *Proceedings of the Congress on Evolutionary Computation*, vol. 2. COEX, World Trade Center, 159 Samseong-dong, Gangnam-gu, Seoul, Korea: IEEE Press, 27-30 May 2001, pp. 1070–1077. [Online]. Available: http://goanna.cs.rmit.edu.au/ toml/cec2001.ps
[22] A. Teller and M. Veloso, "A controlled experiment : Evolution for learning difficult image classification," in *Proceedings of the 7th Portuguese Conference on Artificial Intelligence*, ser. LNAI, C. Pinto-Ferreira and N. J. Mamede, Eds., vol. 990. Berlin: Springer Verlag, 3–6 Oct. 1995, pp. 165–176.
[23] ——, "PADO: Learning tree structured algorithms for orchestration into an object recognition system," Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, Tech. Rep. CMU-CS-95-101, 1995.
[24] M. Zhang and V. Ciesielski, "Genetic programming for multiple class object detection," in *Proceedings of the 12th Australian Joint Conference on Artificial Intelligence (AI'99)*, N. Foo, Ed. Sydney, Australia: Springer-Verlag Berlin Heidelberg, December 1999, pp. 180–192, lecture Notes in Artificial Intelligence (LNAI Volume 1747).
[25] W. Kantschik, P. Dittrich, M. Brameier, and W. Banzhaf, "Metaevolution in graph GP," in *Genetic Programming, Proceedings of EuroGP'99*, ser. LNCS, R. Poli, P. Nordin, W. B. Langdon, and T. C. Fogarty, Eds., vol. 1598. Goteborg, Sweden: Springer-Verlag, 26-27 May 1999, pp. 15–28.
[26] W. Kantschik and W. Banzhaf, "Linear-graph GP—A new GP structure," in *Proceedings of the 4th European Conference on Genetic Programming, EuroGP 2002*, E. Lutton, J. A. Foster, J. Miller, C. Ryan, and A. G. B. Tettamanzi, Eds., vol. 2278. Kinsale, Ireland: Springer-Verlag, 3-5 2002, pp. 83–92. [Online]. Available: citeseer.nj.nec.com/kantschik02lineargraph.html
[27] P. A. Whigham, "Grammatically-based genetic programming," in *Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications*, J. P. Rosca, Ed., Tahoe City, California, USA, 9 July 1995, pp. 33–41. [Online]. Available: http://citeseer.ist.psu.edu/whigham95grammaticallybased.html
[28] D. Andre, "Automatically defined features: The simultaneous evolution of 2-dimensional feature detectors and an algorithm for using them," in *Advances in Genetic Programming*, K. E. Kinnear, Ed. MIT Press, 1994, pp. 477–494.
[29] G. Robinson and P. McIlroy, "Exploring some commercial applications of genetic programming," in *Evolutionary Computation, Volume 993, Lecture Note in Computer Science*, T. C. Fogarty, Ed. Springer-Verlag, 1995.
[30] J. F. Winkeler and B. S. Manjunath, "Genetic programming for object detection," in *Genetic Programming 1997: Proceedings of the Second Annual Conference*, J. R. Koza, K. Deb, M. Dorigo, D. B. Fogel, M. Garzon, H. Iba, and R. L. Riolo, Eds. Stanford University, CA, USA: Morgan Kaufmann, 13-16 July 1997, pp. 330–335.
[31] D. Howard, S. C. Roberts, and C. Ryan, "The boru data crawler for object detection tasks in machine vision," in *Applications of Evolutionary Computing, Proceedings of EvoWorkshops2002: EvoCOP, EvoIASP, EvoSTim*, ser. LNCS, S. Cagnoni, J. Gottlieb, E. Hart, M. Middendorf, and G. Raidl, Eds., vol. 2279. Kinsale, Ireland: Springer-Verlag, 3-4 Apr. 2002, pp. 220–230.
[32] C. T. M. Graae, P. Nordin, and M. Nordahl, "Stereoscopic vision for a humanoid robot using genetic programming," in *Real-World Applications of Evolutionary Computing*, ser. LNCS, S. Cagnoni, R. Poli, G. D. Smith, D. Corne, M. Oates, E. Hart, P. L. Lanzi, E. J. Willem, Y. Li, B. Paechter, and T. C. Fogarty, Eds., vol. 1803. Edinburgh: Springer-Verlag, 17 Apr. 2000, pp. 12–21.
[33] D. Howard, S. C. Roberts, and C. Ryan, "Evolution of an object detection ant for image analysis," in *2001 Genetic and Evolutionary Computation Conference Late Breaking Papers*, E. D. Goodman, Ed., San Francisco, California, USA, 9-11 July 2001, pp. 168–175.
[34] F. Lindblad, P. Nordin, and K. Wolff, "Evolving 3d model interpretation of images using graphics hardware," in *Proceedings of the 2002 IEEE Congress on Evolutionary Computation, CEC2002*, Honolulu, Hawaii, 2002.
[35] P. Nordin and W. Banzhaf, "Programmatic compression of images and sound," in *Genetic Programming 1996: Proceedings of the First Annual Conference*, J. R. Koza, D. E. Goldberg, D. B. Fogel, and R. L. Riolo, Eds. Stanford University, CA, USA: MIT Press, 1996, pp. 345–350.
[36] R. Poli, "Genetic programming for feature detection and image segmentation," in *Evolutionary Computing*, ser. Lecture Notes in Computer Science, T. C. Fogarty, Ed. University of Sussex, UK: Springer-Verlag, 1-2 Apr. 1996, no. 1143, pp. 110–125.

APPENDIX
GENETIC PROGRAMMING BASICS

Constructing a GP system involves making design decisions for a number of elements of the GP system, including the representation of programs, the construction of an initial population of programs, the evaluation of programs and the construction of new population of programs. This appendix briefly describes the basic aspects of GP, including program representation, program generation, the primitive set, the fitness function, the selection mechanism, the genetic operators and the overall GP algorithm. More detailed description on GP can be seen from [6], [5].

A. Program Representation

Much of the GP work was done using LISP or LISP-like representations of the programs. A sample computer program for the algebraic equation $(x - 1) - x^3$ can be represented in LISP as the S-expression `(- (- x 1) (* x (* x x)))`. The tree representation is shown in figure 9.

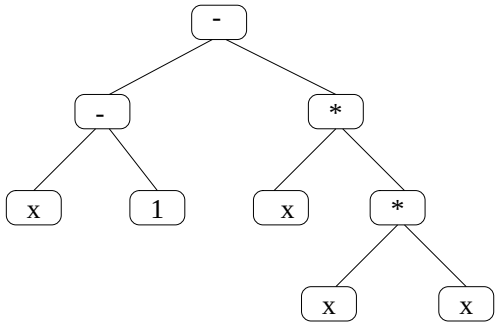


Fig. 9. A simple tree representation for a sample LISP program.

The programs are constructed from *a terminal set* and *a function set* which vary according to the problem domain. Terminals and functions are also called *primitives*, and the terminal set and the function set are combined to form a *primitive set*.

Functions in a function set form the internal nodes of the tree representation of a program. In general, there are two kinds of functions used in genetic programming. The first class refers to standard functions, such as the four arithmetic

operations. The second class comprises specific functions which vary with the problem domain.

Terminals have no arguments and form the leaves of the parse tree. Typically, terminals represent the inputs to the GP program, the constants supplied to the GP program, or zero-argument functions with side-effects executed by the GP program [5]. In any case, a terminal returns an actual numeric value without having to take an input.

B. Program Generation

There are several ways of generating programs to initialise a GP population, including *full*, *grow* and *ramped half-and-half* [6]. In the full method, functions are selected as the (internal) nodes of the program until a given depth of the program tree is reached. Then terminals are selected to form the leaf nodes. This ensures that full, entirely balanced trees are constructed. When the grow method is used, nodes are selected from either functions or terminals. If a terminal is selected, the generation process is terminated for the branch and moves on to the next non-terminal branch in the tree. In the ramped half-and-half method, both the full and grow methods are combined. Half of the programs generated for each depth value are created by using the grow method and the other half using the full method.

C. Fitness Function

Fitness is the measure of how well a program has learnt to predict the output from the input during simulated evolution. The fitness of a program generated by the evolutionary process is computed according to the fitness function. The fitness function should be designed to give graded and continuous feedback about how well a program in a population performs on the training set.

D. Selection Mechanism

The selection mechanism determines which evolved program will be used for the genetic operators to produce new individuals for the next generation during the evolutionary process. Two of the most commonly used selection methods are *proportional selection* and *tournament selection*.

In the proportional selection method [6], an individual in a population will be selected according to the proportion of its own fitness to the total sum of the fitness of all the individuals in the population. Programs with low fitness scores would have a low probability of having any genetic operators applied to them and so would most likely be removed from the population. Programs which perform particularly well in an environment will have a very high probability of being selected.

The tournament selection method [5] is based on competition within only a subset of the population, rather than the whole population. A number of programs are selected randomly according to the tournament size and a selective competition takes place. The better individuals in the tournament are allowed to replace the worse individuals. In the smallest possible tournament, two individuals can compete. The winner is allowed to reproduce with mutation and the result is returned to the population, replacing the loser of the tournament.

E. Genetic Operators

There are three fundamental genetic operators: reproduction, mutation and crossover.

Reproduction is the basic engine of Darwinian theory [6], which involves just simply copying the selected program from the current generation to the new generation. This allows good programs to survive during evolution.

Mutation operates only on a single selected program and introduces new genetic code in the new generation. This operator removes a random subtree of a selected program, then puts a new subtree in the same place. The goal here is to keep the diversity of the population in evolution.

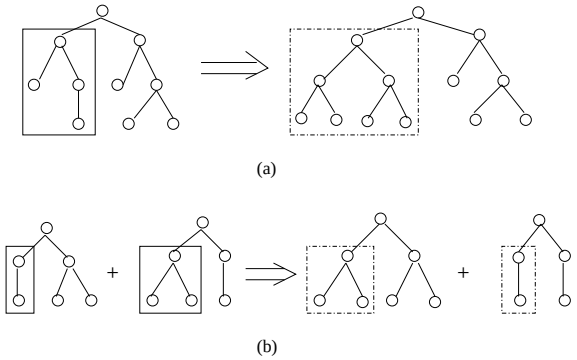


Fig. 10. Effect of genetic operators in genetic programming. (a) Mutation in GP: Replaces a random subtree; (b) Crossover in GP: Swaps two random subtrees.

Crossover takes advantage of different selected programs within a population, attempting to integrate the useful attributes from them. The crossover operator combines the genetic material of the two selected parents by swapping a subtree of one parent with a subtree of the other, and introducing two newly formed programs into the population in the next generation.

F. The GP Algorithm

The learning/evolutionary process of the GP algorithm is summarised as follows:

1) Initialise the population.
2) Repeat until a termination criterion is satisfied:
   - 2.1 Evaluate the individual programs in the current population. Assign a fitness to each program.
   - 2.2 Until the new population is fully created, repeat the following:
     - Select programs in the current generation.
     - Perform genetic operators on the selected programs.
     - Insert the result of the genetic operations into the new generation.
3) Present the best individual in the population as the output — the learned/evolved genetic program.

An Effective Tree-Based Algorithm for Ordinal Regression

Fen Xia, Wensheng Zhang, and Jue Wang, *Senior Member, IEEE*

***Abstract*— Recently ordinal regression has attracted much interest in machine learning. The goal of ordinal regression is to assign each instance a rank, which should be as close as possible to its true rank. We propose an effective tree-based algorithm, called Ranking Tree, for ordinal regression. The main advantage of Ranking Tree is that it can group samples with closer ranks together in the process of tree learning. This approach is compared with original decision tree. Experiments on some synthetic and real-world datasets show that Ranking Tree outperforms original decision tree in terms of speed and accuracy as well as robustness.**

***Index Terms*— Machine learning, ranking, decision tree, splitting rule.**

I. Introduction

CONSIDER the following stamp-rating scenario. As a stamp collector, Jack has already collected a lot of stamps in the past few years. However, he is still looking for new stamps. Whenever he gets a stamp, he would need to rate the stamp based on a 1-5 scale, with 5 representing the most valuable collection.

Jack's rating problem can be modeled as a supervised inductive learning task. Two most popular supervised inductive learning methods are classification and regression. In classification, unknown labels are estimated from a set of finite, *unordered* categories. In regression, numeric outputs take continuous values. However, Jack's rating problem cannot be directly solved by either of these two methods because labels in this case are chosen from a set of finite, *ordered* ratings. In the literature, Jack's problem is one that predicts instances of ordinal scale, i.e., the so-called ordinal regression [1].

Applications of ordinal regression frequently arise from domains where human-generated data play an important role. Examples of these domains include information retrieval, collaborative filtering, medicine, and psychology. When people assess objects of interest in these domains (e.g., in terms of their correctness, quality, or any other characteristics), they often resort to subjective evaluation and provide rating information that is typically imprecise. Also, rating results or scores given by different persons are usually not comparable. Therefore, ordinal labels are preferred to continuous scores. In practice, ordinal labels typically correspond to linguistic terms such as "very bad", "good", "very good".

Several approaches have been developed in the machine learning literature to deal with ordinal regression. One obvious idea is to convert the ordinal regression to the regular regression problem. For instance, [2] investigated the use of a regression tree learner by mapping rating results to real values. However, determining an appropriate mapping is often difficult because the true, underlying metric among the ordinal scales is unknown for most tasks. As a result, these regression algorithms are more sensitive to the representation of the ranks rather than the ordinal relationships. Another idea is to convert the ordinal regression to a multi-class classification problem [3]. In these approaches, the ordinal regression problems are converted into nested binary classification problems and the results of these binary classifications are combined to produce for rating prediction. It is also possible formulate the ordinal regression as a large augmented binary classification problem. [1] applied the principle of structural risk minimization to ordinal regression, leading to a new distribution-independent learning algorithm based on a loss function defined on pair items of different ranks. [7] considers general ranking problems in the form of preference judgments and presents a complexity gap between classification and ranking. [8] presents a formal framework for the general ranking problem in the form of preference judgments. However, these approach are time consuming as they operate on pre-processed datasets whose size is quadratic of that of the original dataset. As for on-line learning, [4] and [5] operate directly on ranks by associating each rank with a distinct sub-interval on the real line and those intervals are adapted in the process of learning. [6] generalizes the approach of [4] and [5] to deal with the ranking and re-ranking problem in natural language processing. The ranking algorithm in [6] searches dynamically for pairs of inconsistent objects with different margins and uses them to update the weight vector. Other methods have also been proposed. [9] presents a probabilistic kernel approach to ordinal regression based on Gaussian processes. [10] generalizes the formulation of support vector machines to ordinal regression. [11] uses gradient descent method for learning ranking functions based on the pairs items.

In this paper we develop an alternative approach that uses a decision tree [12], [13], [14] with a suitable splitting rule

This work was supported in part by the National Basic Research Program of China (2004CB318103), National Science Foundation of China (60033020), National Science Foundation (60575001) of China, and Overseas Outstanding Talent Research Program of Chinese Academy of Sciences(06S3011S01).

Fen Xia is with the Key Laboratory of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, P.R. China, and also with the Graduate School, Chinese Academy of Sciences, Beijing, P.R. China (e-mail: fen.xia@ia.ac.cn).

Wensheng Zhang is with the Key Laboratory of complex system and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, P.R China (email: wensheng.zhang@mail.ia.ac.cn).

Jue Wang is with the Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, P.R China(email: jue.wang@mail.ia.ac.cn).

for ordinal regression. As a widely-used data mining and machine learning tool [17], [18], decision trees can achieve good prediction accuracy while producing an easy-to-interpret rule. It can accept continuous, discrete and categorical inputs. It is invariant under strictly monotone transformations of the individual inputs and performs internal feature selection as an integral part of the procedure. Therefore, it is quite desirable to use decision tree for ordinal regression. To our best knowledge, the use of tree learners in ordinal regression is largely under-explored. [2] investigated the indirect use of a regression tree learner to tackle ordinal regression problems. However, their method requires a proper mapping function, which in many cases can only be heuristically defined through trials-and-errors. Another possible use of the tree learner in ordinal regression is to formulate the ordinal regression problem as a multi-class classification problem. As is well known, splitting rule is a growth strategy which guides the learning of the tree. A major problem with this method is that the splitting rule in classification does not take the ordinal relationship into account. The key technical challenge with developing a tree-based ordinal regression method, in our view, is the development of a proper splitting rule that can make use of the ordinal relationship.

The splitting rule is based on the impurity measure of a set. Thus, development of a proper splitting rule is equal to seek a proper impurity measure. We present a new impurity measure motivated by the following intuition. The impurity of a set can be decided by the deviation of sample ratings in the set. A pair of irrelevant items should cause more impurity than a relevant or possibly relevant pair. Likewise, the more pairs with different ratings in a set, the more impure the set will be. We formalize this intuition by developing a new impurity measure on a set.

The reported research is based on this new impurity measure. We use it to construct the splitting rule for the ordinal regression problem. Based on the splitting rule, we train a decision tree, called Ranking Tree. This method is compared with the original classification tree using some synthetic and real-world datasets. Experiments show that Ranking Tree outperforms the classification tree in terms of speed and accuracy as well as robustness.

The remainder of this paper is organized as follows. In Section 2, we present two impurity measures: the gini impurity, a popular measure widely used in the classification tree literature and the base of comparison for our measure; the ranking impurity, our measure proposed in this paper; Section 3 presents a detailed analysis of these two measures. In Section 4, we experimentally compare the Ranking Tree with the classification tree and summarize the results. In Section 5, we conclude the paper and point some possible future research directions.

II. Two Impurity Measures

A. The Gini Impurity

One of the most commonly used impurity measures in classification problems is the gini impurity, defined as follows:

Definition 1: Given a sample set T, let $p_i = p(i|T)$ be the relative proportion of class i samples in the set T, where $i \in \{1, \ldots, k\}$ is the class label; the gini impurity (also known as the gini index) is defined as

$$I_{gini}(T) = \sum_i \sum_j {}_{j \neq i} p_i p_j$$

There are two interpretations of the gini impurity. If a sample belongs to class i with probability p_i, the loss of misclassifying it would be $p_i \sum_{j \neq i} p_j$. Therefore, the expected loss on all classes due to misclassifications is given by $\sum_i \sum_j {}_{j \neq i} p_i p_j$. In the second interpretation, if each sample is coded as 1 for the class i with probability p_i and zero otherwise, the variance of this code variable is $p_i(1 - p_i)$. Summing these variances over all classes produces the gini impurity.

With the impurity measure, sets can be compared. Also, the split associated with sets can be compared. A split is to divide a set T into two sets T_L and T_R, corresponding to the left child and the right child of T respectively. The splitting rule of gini impurity is to find the best split, which is the one that maximizes the quantity defined as

$$\Delta I = I_{gini}(T) - I_{gini}(T_L)p(T_L) - I_{gini}(T_R)p(T_R)$$

This objective can be interpreted as to minimize error of random rule in child nodes.

The gini index is well suitable for standard classification tasks. However, in ordinal regression, the gini index ignores the ordinal relationship among the class labels in that all class labels are treated equally. Furthermore, consider the first interpretation discussed above. Misclassifying a sample from class i to every other class produces an equal portion of loss. This is problematic in ordinal regression because ranking an item further away from its actual rank would be more harmful.

B. The Ranking Impurity

We now present our new impurity measure named ranking impurity.

Definition 2: Given a sample set T labeled by a totally ordered set $L = \{L_1, \ldots, L_k\}$, let $N_i(T)$ be the number of elements in T that have label L_i; the ranking impurity is given by:

$$I_{rank}(T) = \sum_{j=1}^{k} \sum_{i=1}^{j} (j - i) N_j(T) N_i(T)$$

The ranking impurity can be interpreted as the maximum potential number of miss-ranked pairs in the set. Imagine a rater who always makes a mistake when he evaluates a pair of objects. For example, if one sample a_1 belongs to rating L_1, and another sample a_2 belongs to rating L_2, he will always give a wrong order and rank a_1 after a_2. To measure the extent of such mistakes, we weigh the pair by the difference of the ratings, that is, $L_2 - L_1$. Since a set can be decomposed into many pairs, the maximum mistakes that the rater will make are our ranking impurity.

The splitting rule of the ranking impurity is then to find the best split, which is the one that maximizes the quantity defined as

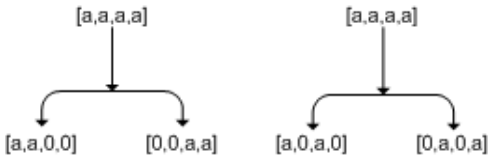


Fig. 1. Two splits of the decision tree.

$$\Delta I = I_{rank}(T) - I_{rank}(T_L) - I_{rank}(T_R) \qquad (1)$$

The objective can be interpreted as to minimize the maximum potential number of miss-ranked pairs in both T_L and T_R.

It is easy to verify that the ΔI in (1) is positive whenever neither T_L nor T_R is empty. So it prevents the creation of degenerate trees.

Roughly speaking, the ranking impurity emphasizes the role of individual samples while the gini impurity emphasizes the role of the individual classes. Meanwhile, the former takes the order relationship into account while the latter not.

III. Ranking Impurity Based Decision Tree Evaluation

In this section, we analyze the ranking impurity and describe its capacity in expressing ordinal relationships.

Consider for instance the two splits in Fig.1.

In both splits, the parent nodes have four ratings (1, 2, 3, 4 with 1 as the first element) and each rating has the same number of a samples. The split in the left tree sends all samples with rating equal 1 and all samples with rating equal 2 to its left child node. Then the remainder is sent to its right child node. On the other hand, the split in the right tree sends all samples with rating equal 1 and all samples with rating equal 3 to its left child node. Then the others are sent to its right child node.

Now we evaluate these two splits using the gini and ranking impurity measures. The child nodes have the same weighted average gini impurity in both splits. In contrast, the left split leaves a ranking impurity of $2a^2$ while the right split $4a^2$. Therefore, ranking impurity prefers the left split to the right split.

Comparing the two splits, we observe that the samples of closer ratings are bundled together in the left split but not in the right split. We omit the theoretical proof due to the lack of space and state that partitioning with rank impurity can group samples with closer ratings together in each splitting step. Consider a case where there are $N_1(T)$ samples of rating 1, $N_2(T)$ samples of rating 2 and $N_3(T)$ samples of rating 3 at a node T. If $N_1(T)$, $N_2(T)$, and $N_3(T)$ are equal, the split with ranking impurity will never separate out those $N_2(T)$ samples of rating 2. On the other hand, since the split with gini impurity ignores the ordinal relationship and it may separate out the samples of rating 2. If $N_2(T) \leq 2N_1(T)$, or $N_2(T) \leq 2N_3(T)$, then the splitting with ranking impurity will avoid separating out the samples of rating 2.

In ordinal regression, it is important to group the samples with closer ratings together in each splitting step for the following reasons: Firstly, it might lead to a fast error convergence rate measured by the deviation from the true rank in the process of the partitioning. Secondly, it provides a robust method to deal with noises. The ratings given by users often contain noise; for instance, the rater often is unsure about which one of the adjacent ratings to assign. Partitions aimed to preserve the ranking of samples may be less affected by these noises than simple partitioning since they would tend to put samples of adjacent ratings together in one node. Computationally, the two impurity measures share the same goal, that is, making the leaf nodes pure. However, the process of splitting can be very different because of the greedy nature of the tree-based algorithms. We argue that splitting with ranking impurity is more suitable than splitting with gini impurity in ordinal regression. The next section reports experimental findings that support this argument.

IV. Experiments and Discussion

To compare the Ranking Tree algorithm with the classification tree algorithm, we use one synthetic dataset and several real-world datasets. In our experiments the CART decision tree algorithm was used, with the splitting rule specified either by the gini or ranking impurity measure. The implementation of CART was based on the *rpart* package in R, which can be found at *http://www.r-project.org*.

A. Evaluation using a synthetic dataset

We generated a synthetic dataset using the same data generation process as specified in [1], [4], [5]. Firstly, we generated random points according to the uniform distribution on the unit square $[0, 1] \times [0, 1]$. Then we assigned each point with the rank chosen from set $\{1, \ldots, 5\}$ using the following ranking rule, $y = \max_r\{r : 10((x_1 - 0.5)(x_2 - 0.5)) + \varepsilon > b_r\}$ where $b = \{-\infty, -1, -0.1, -0.25, 1\}$ and ϵ was normally distributed with zero mean and standard deviation of 0.125. We used the measure, which quantified the accuracy of predictive ordinal ranks with respect to true ranks, i.e., the average rank loss $\frac{1}{T}\sum_{t=1}^{T} |\widehat{y}_t - y_t|$, where T is the number of samples in the test set.

We used 20 Monte-Carlo trials with $50,000$ training samples and a separate test set of $1,000$ samples to compare the performance of the two algorithms in the large training datasets. Cross-validation was used to choose the depth of the tree. Table I shows the results of Ranking Tree and classification tree.

TABLE I
The average rank loss $\frac{1}{T}\sum_{t=1}^{T} |\widehat{y}_t - y_t|$ with their corresponding 95 percent confidence intervals with the Student's t-distribution produced by separate test samples with different algorithm in the synthetic dataset, where T is the test set size. RT refers to our Ranking Tree. CT refers to the Classification Tree. Depth refers to the depth of the tree.

Algorithm	Rank loss
RT with depth = 9	0.16±0.01
CT with depth = 9	0.17±0.01

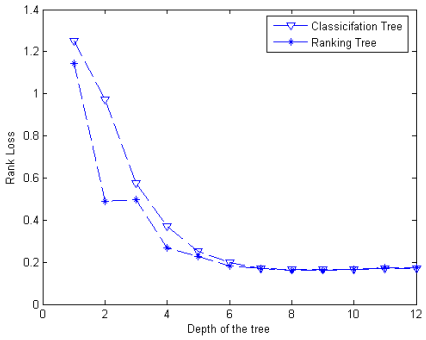


Fig. 2. The average 5-fold cross-validated ranking loss of classification tree and Ranking Tree, with respect to the depth of the tree.

From Table I, we note that the performances of the RT and CT algorithms are very close. It is shown that given enough training samples the RT and CT algorithms can achieve almost the same overall performance.

To compare the convergence rates of the two algorithms, we used $50,000$ training samples and recorded their 5-fold cross-validation results in the process of partitioning. Fig. 2 shows the results of the two algorithms with respect to the depth of the tree. Ranking Tree exhibits a much faster convergence rate than classification tree. This observation supports our analysis, which predicted that Ranking Tree would create better partitions than the classification tree. We also notice the closely-matched performance of classification tree and Ranking Tree as the tree depth increases. We suspect that this is due to the fact that both algorithms are able to find a partition that every node in the tree is very " pure ", resulting in similar performance.

To model noises in the data, we defined a noise level σ , and assumed that each rating could be "misranked" to its adjacent ratings with probability σ.

We used 20 Monte-Carlo trials to test the two algorithms in different size of the training samples. All the results were produced by 5-fold cross validation and were shown in Fig. 3.

From Fig. 3 we can see that Ranking Tree algorithm achieves lower rank loss and delivers much tighter confidence intervals than the classification tree algorithm in all conditions, especially when the size of training samples is small and the noise level is high. This supports our claim that the Ranking Tree is more robust than the classification tree algorithm.

B. Ranking with real-world collaborative filtering datasets

For testing purposes, we chose two real-world collaborative filtering datasets; both of them were used for ordinal regression research [5]: Cystic Fibrosis [15] and MovieLens dataset [16]. The original datasets are composed of the items where each entry is given by a query-document-rating triple. We constructed the dataset in the following way. We randomly chose a target rank y_t on one item and then used the remaining ratings as the dimensions of the instance vector x_t. The detailed experimental setup for each dataset is described below.

The Cystic Fibrosis dataset is a set of 100 queries with the respective relevant documents. There are $1,239$ documents published from 1974 to 1979 discussing Cystic Fibrosis. In each query-document pair, there are three ratings of highly relevant, marginally relevant and not relevant, which we used the ranks of 3, 2, 1 to represent respectively. There are four ratings for each query-document pair. In the end, we have three dimensions of the feature vector and a target rank. The training set and test set sizes were $4,247$ and 572, respectively.

The MovieLens dataset consists of $100,000$ ratings $(1-5)$ from 943 users on $1,682$ movies, with each user rating at least 20 movies. We considered only those people who had rated over 300 movies. There are 54 persons in total in this category; as such the dimension of the instance vectors is 53. Firstly, we randomly chose a target person from the 54 people. Then we looked for the first 300 movies rated by him and formed an instance by using his ratings as the target rank. While doing so, the ratings from the remaining 53 people about the same movie forms the feature vector. If one of those 53 people had not seen a selected movie, we assigned rank 3 to that movie for the people. In this set of experiments, 210 random items were selected to form the training set and the remaining 90 movies served as the test set.

We tested our Ranking Tree and classification tree in the two collaborative filtering datasets. As in the case of the synthetic dataset, We also used the averaged rank loss $\frac{1}{T}\sum_{t=1}^{T}|\widehat{y}_t - y_t|$, where T is the test set size. The results were averaged on 500 Monte-Carlo trials and are given in Table II.

TABLE II

TEST SET PERFORMANCE ON COLLABORATIVE FILTERING. THE PERFORMANCE MEASURE IS THE AVERAGED RANK LOSS $\frac{1}{T}\sum_{t=1}^{T}|\widehat{y}_t - y_t|$, WHERE T IS THE TEST SET SIZE. THE RESULT IS REPRESENTED WITH THEIR CORRESPONDING 95 PERCENT CONFIDENCE INTERVALS WITH THE STUDENT'S T-DISTRIBUTION.

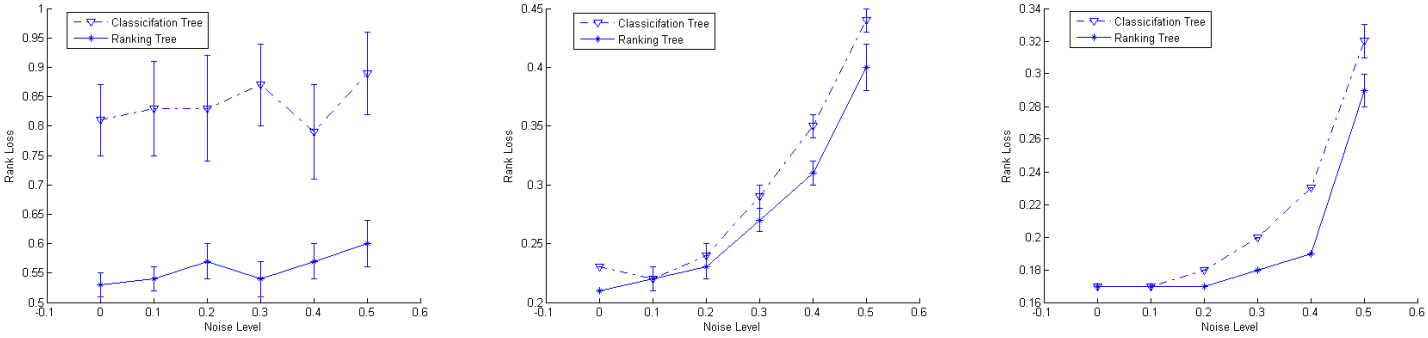
Algorithm	Cystic Fibrosis	MovieLens
RT	0.27±0.00 (Depth = 6)	0.79±0.02 (Depth = 2)
CT	0.39±0.00 (Depth = 4)	0.80±0.02 (Depth = 1)

From Table II we observe that on the Cystic Fibrosis dataset Ranking Tree significantly outperforms the classification tree. Interestingly, on the MovieLens dataset both classification tree and Ranking Tree prefer trees with fewer nodes. Also, it turns out that stumps (trees with depth 1) perform rather well on that dataset. This might imply that if given enough number of recommenders, one's recommendation would nearly always be similar to some other recommender's.

V. CONCLUSION

In this paper, we have presented an effective approach to ordinal regression based on decision tree embedding a new splitting rule based on rank impurity. We have experimentally validated this approach, demonstrating its performance and robustness, relative to an existing approach based on the gini impurity.

Decision tree algorithms have many practical merits. They can handle continuous, discrete and categorical features, fill missing values and select relevant features to produce simple rules. By applying the ranking impurity metric on decision tree, Ranking Tree preserves those merits.



(a) (b) (c)

Fig. 3. Learning curves for Classification Tree (dashed-dotted line) and Ranking Tree (solid line) if we measure average rank loss. The error bars indicate the 95 confidence intervals of the estimated rank loss. (a) The size of training set is 100; (b) The size of training set is 1000; (c) The size of training set is 10000.

Decision trees are known to be instable. Techniques like bagging and boosting can be applied to greatly reduce the instability of decision trees. However, as these algorithms were originally defined on the classification or regression case, extending them to the ordinal regression problem will be a challenge. Our current research is addressing this challenge.

The reported work deals with totally ordered ratings only. In many applications, the sample set might have several subsets, with a different order defined on each one. We are working on investigating whether Ranking Trees can be extended to tackle these generalized ordinal regression problems.

REFERENCES

[1] R. Herbrich, T. Graepel, and K. Obermayer “Large margin rank boundaries for ordinal regression,” *Advance in Large Margin Classifiers*, pp 115–132, 2000.

[2] S. Kramer, G. Widmer, B. Pfahringer, and M. DeGroeve, “Prediction of ordinal classes using regression trees,” *Fundamenta Informaticae, 47*, pp. 1–13, 2001

[3] E. Frank and M. Hall, “A simple approach to ordinal classification” *Proceedings of the European Conference on Machine Learning*, pp. 145–165, 2001.

[4] K. Crammer and Y. Singer, “ Pranking with ranking,” *Proceedings of the conference on Neural information Processing Systems(NIPS)*, 2001.

[5] Edward F. Harrington, ”Online Ranking/Collaborative filtering using the Perceptron Algorithm,” *In Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, Washington DC, 2003.

[6] Libin Shen and Aravind K. Joshi, ”Ranking and Reranking with Perceptron,” *Maching Learning*, vol.60, pp. 73-96, 2005.

[7] W.W. Cohen, R.E. Schapire, and Y. Singer, ” Learning to order things,” *Journal of Artificial Intelligence Research(JAIR)*, 10:243-270, 1999.

[8] Y. Freund, R. Iyer, R.E. Schapire, and Y. Singer, ”An efficient boosting algorithm for combining preferences,” *Journal of Machine Learning Research*, 4:933-969, 2003.

[9] W. Chu and Z. Ghahramani, ”Gaussian processes for ordinal regression,” *Journal of Machine Learning Research*, 6:1019-1041, 2005.

[10] A. Shashua and A Levin, “Ranking with Large Margin Principle: Two Approaches,” *Proceedings of the conference on Neural information Processing Systems. (NIPS)*14* , 2003.

[11] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton and Greg Hullender, ”Learning to Ranking using Gradient Descent,” *In Proceedings of the 22nd International Conference on Maching Learning(ICML-2005)*, Bonn, Germany, 2005.

[12] Leo Breiman, Jerome H. Friedman, Richard A. Olshen and Charles J. Stone, *Classification and Regression Trees*, Wadsworth, 1984.

[13] J. R. Quinlan, ”Induction of decision trees”, *Machine Learning*, 1:81-106, 1986.

[14] J. R. Quinlan, ”C4.5: Programs for Machine Learning”, Morgan Kaufmann, 1993.

[15] Shaw, W.M, Wood, J.B, Wood, R.E and Tibbo, H.R, *The Cystic Fibrosis Database: Content and Research Opportunities. LISR 13*, pp. 347-366, 1991.

[16] GroupLens Research Project, ”MovieLens data sets,” *http://www.grouplens.org/data/*

[17] Shichao Zhang, Xindong Wu and Chengqi Zhang ”Multi-Database Mining”, *The IEEE Intelligent Informatics Bulletin*, Vol.2, No.1, June, 2003.

[18] Xindong Wu, ”Data Mining: An AI Perspective”, *The IEEE Intelligent Informatics Bulletin*, Vol.4, No.2, Dec, 2004.

On Intelligence

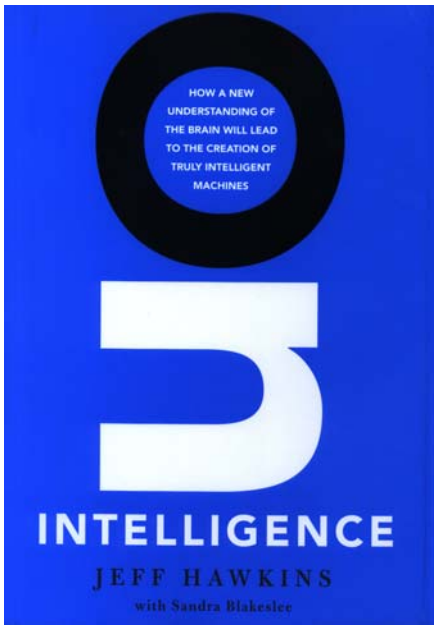
BY JEFF HAWKINS, HENRY HOLT & CO, NY, 2004. ISBN 0-8050-7456-2

REVIEWED BY MIKE HOWARD[1]

It naturally makes researchers bristle when a new guy comes along and makes grand pronouncements that everyone else has missed the big idea. Stephan Wolfram encountered some of that with his 1192 page tome, "A New Kind of Science" where he argued that cellular automata do much better at simulating complex phenomena than more sophisticated models. It helps things not a bit when the new guy struck it rich by inventing a cool tech gizmo and now wants to do "real science." Jeff Hawkins is such a man, asserting that he has studied the esoteric findings of brain researchers, and has new insight about how it all fits together.

Hawkins' credentials as a creative person with plenty of his own natural intelligence are stunning. Every review of Jeff Hawkins' new book begins with a note that he is the brilliant entrepreneur and computer expert who founded Palm Computing and Handspring. He invented the PalmPilot, the Treo smart phone and other gadgets. One of his gifts is being able to make technology simple and accessible, and that goes for his writing as well.

In "On Intelligence," Jeff Hawkins presents a new theory about how the brain works and how we can finally build "intelligent" machines. Of course, machine intelligence has been a goal of computer science for decades. He discusses the success of Artificial Intelligence in the 60's and 70's that led to an irrational exuberance when the popular press started hyping the possibility of creating artificial brains. In the 90's the bubble burst when the fickle press discovered those wild expectations were unrealistic, and researchers nearly had to stop using the term "AI".



Progress continued to be made in the foundations of AI of course, and it was good for the field when the hype died down. Fuzzy logic chips appeared in washing machines, automated planning systems controlled NASA space probes including the Mars Rover, and expert systems are used by banks to decide who to loan money to. Neural networks have had success in pattern recognition. But although great strides have been made in machine learning, and Moore's Law has made many AI algorithms practical, it is clear that AI alone will not result in a robot that can do your errands for you or babysit your children.

Hawkins points out that the Turing Test is not, after all, a good indicator of intelligence. John Searle, a philosopher and cognitive scientist who created a thought experiment called the "Chinese Room," indicated that while a computing device could indeed reply to questions in such a way that made it indistinguishable from a human being, it did not understand the conversation and there was no meaning attached to its replies. Although Searle's experiment is controversial, it suggests that Turing's test is faulty and misleading.

Hawkins believes that our best bet for learning to build truly intelligent machines is to learn how the brain actually operates. In particular, it is the neocortex, the center of higher thought, which is the focus of his attention. But he also implies that neuroscientists are lost in the complexity of mapping out neural pathways, and are not coming up with compelling overarching theories that begin to explain how we think and learn.

Hawkins believes there is enough evidence now to posit a common cortical algorithm, as first proposed by Vernon Mountcastle, a neuroscientist at Johns Hopkins, in 1978. The algorithm is hierarchical, with lower layers encoding data from a sensory organ, but higher layers dealing with abstract signals that bear little resemblance to the sensory signals. Hawkins asserts that brain researchers got sidetracked partly due to the experimental difficulty of taking measurements. The standard approach is to present a static sensory stimulus and take readings of resulting cortex activity. It is too difficult to work with dynamically changing stimuli, so researchers have missed a point that Hawkins believes is crucial: the brain can only perceive dynamic stimuli.

Since the author believes that brain research is mired in complexity, a higher level theory is needed to provide a top-down pressure to guide the field. By the way, he makes no mention of Minsky's "Society of Mind," a metaphorical / philosophical thought experiment less constrained by brain research than what Hawkins had in mind.

Hawkins' theory, called "Memory Prediction Framework," defines *Intelligence* as "the capacity of the brain

[1] mhoward@hrl.com, HRL Laboratories, LLC

to predict the future by analogy to the past." According to him, there are four key attributes of neocortical memory that differ from computer memory:

- All memories are inherently sequential.
- Memory is auto-associative; a partial memory can be used to retrieve the full memory.
- Memories are stored in invariant representations.
- Patterns are stored in a hierarchy.

Support for the theory is most concretely expressed in chapter six, the meatiest part of the book. This is where the author describes in some detail his vision of how the neural circuitry in the layers of cortex works. The description is compelling, but takes more work to follow than the other chapters.

Chapter six ends with several fascinating observations that are built on top of the neural circuitry described earlier. It emphasizes that perception and behaviour are highly interdependent because they both originate in a detail-invariant representation that is then transmitted through both motor and sensory cortex. Also, although many researchers have discounted it, Hawkins argues that feedback and the importance of distant synapses in cortex is essential to explain the Memory Prediction Framework theory, and should be reconsidered. The theory includes the broad principles of how hierarchical learning of sequences explains how children first learn letters, then words, phrases and finally sentences, and as adults we can speed-read without needing to study every letter. The author believes that the memory of sequences re-forms lower and lower in cortex, allowing higher layers to learn more complex patterns. Finally, the hippocampus is briefly described as logically residing at the top of the cortical hierarchy: the short-term repository of new memories.

An impressive result of the speculations in chapter six is a list in the appendix of 11 specific, testable predictions made by the theory, which is an invitation to brain researchers. And Hawkins founded a company, Numenta, to develop the Hierarchical Temporal Memory concept based on the theory.

Chapter six also hints at how daydreaming or imagining occurs, when predictions from layer 6 of a cortical column are fed back to layer 4 of the same column. Cortical modeller Stephan Grossberg calls this "folded feedback". In chapter seven the book expands on philosophical speculation about the origin of consciousness and creativity that arise from the Memory Prediction Framework theory. Creativity is defined here as "making predictions by analogy". As the author says, there is a continuum of creativity, from mundane extrapolations from learned sequences in sensory cortex to rare acts of genius. But they have a common origin. This is how a piano player can quickly figure out how to play simple melodies on a vibraphone, or a customer in a strange restaurant can figure out that there is probably a restroom in the back. Creativity is so pervasive that we hardly label it as such, unless it violates our predictions like an unusual work of art. There are practical suggestions in this section for how to train oneself to be more creative, and an interesting story of how Hawkins conceived the handwriting recognition system, Graffiti.

Chapter seven ends in speculation about the nature of consciousness, imagination and reality in response to the inevitable questions to which this type of work gives rise. A review on the Amazon website by Dr. Jonathan Dolhenty takes issue with what he describes as "plain old-fashioned metaphysical materialism and, probably, old-school psychological behaviourism," which are largely discounted theories today. Dolhenty is a philosopher who thinks human intellect at the higher abstract and conceptual levels cannot be described by such a simple extrapolation of the Memory Prediction Framework. But this reviewer found the connections made between brain theory and "mind" reassuring. Leave it to others to build on this foundation. In fact, Hawkins does hint at a broader source of the mind in chapter seven, where he says that it is influenced by the emotional systems of the old brain and by the complexity of the human body.

The last chapter in the book contains another vision, of how intelligent machines might be built in the future. This is back into the Popular Science mode. Unlike many current roboticists who believe humanoid robots will be needed to interact with humans, Hawkins believes humanoid form is pointless and impractical. He advocates working from inside out, by building sensing mechanisms and attaching them to a hierarchical memory system that works on cortex principles. Then by training the system he believes it will develop its own representations of the world. This system can be built into any sort of machine, and the sensors can be distributed if desirable.

The technical challenges of building an intelligent machine include *capacity*, which by analogy to the brain, at 2 bits per synapse, would require 8 trillion bytes of memory or about 80 hard drives. *Connectivity* is a larger problem, since it would be impossible to provide dedicated connections. Hawkins believes the answer would be some sort of shared connections, like in today's phone network, but this is still a challenge.

As an aside, there is no mention of the Cyc project, which has been working since 1984 to build a mammoth semantic knowledge base. But unlike the automatically learned representations in Hawkins' proposed artificial brain, the ones in Cyc are hand-input in a preconceived structure as a vast quantity of terms related by assertions.

The last chapter ends with a very positive view of the potential of intelligent machines to solve problems humans cannot, because they can be equipped with custom senses, immense memory, and even be networked to form hierarchies of intelligent machines. Hawkins believes that intelligent machines will be a hot topic in the next ten years. It is easy to get caught up in his excitement.

Related Websites:
http://www.onintelligence.org/
http://redwood.berkeley.edu/
http://www.numenta.com/

RELATED CONFERENCES, CALL FOR PAPERS/PARTICIPANTS

TCII Sponsored Conferences

AWIC'07
The Fifth Atlantic Web Intelligence Conference
Fontainebleau, France
June 27-29, 2007
http://www.awic2007.net/

The 5th Atlantic Web Intelligence Conference (Spain - 2003, Mexico - 2004, Poland - 2005, Israel - 2006) brings together scientists, engineers, computer users, and students to exchange and share their experiences, new ideas, and research results about all aspects (theory, applications and tools) of intelligent methods applied to Web based systems, and to discuss the practical challenges encountered and the solutions adopted.

The conference will cover a broad set of intelligent methods, with particular emphasis on soft computing. Methods such as (but not restricted to):
Neural Networks, Fuzzy Logic, Multi valued Logic, Rough Sets, Ontologies, Evolutionary Programming, Intelligent CBR, Genetic Algorithms, Semantic Networks, Intelligent Agents, Reinforcement Learning, Knowledge Management, etc. must be related to applications on the Web like: Web Design, Information Retrieval, Electronic Commerce, Conversational Systems, Recommender Systems, Browsing and Exploration, Adaptive Web, User Profiling/Clustering, E-mail/SMS filtering, Negotiation Systems, Security, Privacy, and Trust, Web-log Mining, etc.

.

WI'07
The 2007 IEEE/WIC/ACM International Conference on Web Intelligence
Silicon Valley, USA
November 2-5, 2007
http://www.cs.sjsu.edu/wi07/wi/

Web Intelligence (WI) is a new paradigm for scientific and research and technological development to explore the fundamental interactions between AI-engineering and advanced Information Technology (AIT) on the next generation of Web systems, services, and etc. Here AI-engineering is a general term that refers to a new area, slightly beyond traditional AI: brain informatics, human level AI, intelligent agents, social network intelligence and classical areas, such as knowledge engineering, representation, planning, and discovery and data mining are examples. AIT includes wireless networks, ubiquitous devices, social networks, and data/knowledge grids. WI research seeks to explore the most critical technology and engineering to bring in the next generation Web systems.

Following the great successes of WI'01, WI'03, WI'04, WI'05 and WI'06, Silicon Valley is proposed as the site for the 2007 IEEE/WIC/ACM International Conference on Web Intelligence (WI'07) to be jointly held with the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology, and we expect that this high tech hub will be able to host WI'07 with great success. WI'07 is sponsored by the IEEE Computer Society Technical Committee on Intelligent Informatics (TCII), the Web Intelligence Consortium (WIC), and ACM-SIGART. Holding WI'07 in the heart of the high tech world will provide special opportunities for collaboration between research scientists and engineers.

WI'07 is planned to provide a leading international forum for researchers and practitioners (1) to present the state-of-the-art of WI technologies; (2) to examine performance characteristics of various approaches in Web-based intelligent information technology; and (3) to cross-fertilize ideas on the development of Web-based intelligent information systems among different domains. By idea-sharing and discussions on the underlying foundations and the enabling technologies of Web intelligence, WI'07 will capture current important developments of new models, new methodologies and new tools for building a variety of embodiments of Web-based intelligent information systems.

IAT'07
The 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology
Silicon Valley, USA
November 2-5, 2007
http://www.cs.sjsu.edu/wi07/iat/

Following the great successes of IAT'01, IAT'03, IAT'04, IAT'05 and expected success of IAT'06, we are excited to propose Silicon Valley as the site for the 2007 IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT'07), to be jointly held with the 2007 IEEE/WIC/ACM International Conference on Web Intelligence (WI'07), and we are confident that this high tech hub will be able to host IAT'07 with great success. IAT'07 is sponsored by the IEEE Computer Society Technical Committee on Intelligent Informatics (TCII), the Web Intelligence Consortium (WIC), and ACM-SIGART. Holding IAT'07 in the heart of the high tech world will provide special opportunities for collaboration between research scientists and engineers.

IAT'07 will provide a leading international forum to bring together researchers and practitioners from diverse fields, such as computer science, information technology, business, education, human factors, systems engineering, and robotics, to (1) examine the design principles and performance characteristics of various approaches in intelligent agent technology, and (2) increase the cross-fertilization of ideas on the development of autonomous agents and multi-agent systems among different domains. By encouraging idea-sharing and discussions on the underlying logical, cognitive, physical, and sociological foundations as well as the enabling technologies of intelligent agents, IAT'07 will foster the development of novel paradigms and advanced solutions in agent-based computing.

ICDM'07
The Seventh IEEE International Conference on Data Mining
Omaha, NE, USA
October 28-31, 2007
http://www.ist.unomaha.edu/icdm2007/

The IEEE International Conference on Data Mining series (ICDM) has established itself as the world's premier research conference in data mining. It provides an international forum for presentation of original research results, as well as exchange and dissemination of innovative, practical development experiences. The conference covers all aspects of data mining,

including algorithms, software and systems, and applications. In addition, ICDM draws researchers and application developers from a wide range of data mining related areas such as statistics, machine learning, pattern recognition, databases and data warehousing, data visualization, knowledge-based systems, and high performance computing. By promoting novel, high quality research findings, and innovative solutions to challenging data mining problems, the conference seeks to continuously advance the state-of-the-art in data mining. Besides the technical program, the conference will feature workshops, tutorials, panels and, new for this year, the ICDM data mining contest.

Related Conferences

AAMAS'07
The Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems
Honolulu, Hawaii USA
May 14-18, 2007
http://www.aamas2007.org/

AAMAS'07 encourages the submission of theoretical, experimental, methodological, and applications papers. Theory papers should make clear the significance and relevance of their results to the AAMAS community. Similarly, applied papers should make clear both their scientific and technical contributions, and are expected to demonstrate a thorough evaluation of their strengths and weaknesses in practice. Papers that address isolated agent capabilities (for example, planning or learning) are discouraged unless they are placed in the overall context of autonomous agent architectures or multiagent system organization and performance. A thorough evaluation is considered an essential component of any submission. Authors are also requested to make clear the implications of any theoretical and empirical results, as well as how their work relates to the state of the art in autonomous agents and multiagent systems research as evidenced in, for example, previous AAMAS conferences. All submissions will be rigorously peer reviewed and evaluated on the basis of the quality of their technical contribution, originality, soundness, significance, presentation, understanding of the state of the art, and overall quality of their technical contribution.

In addition to conventional conference papers, AAMAS'07 will also include a demonstrations track for work focusing on implemented systems, software, or robot prototypes; and an industry track for descriptions of industrial applications of agents. The submission processes for the demonstration and industry tracks will be separate from the main paper submission process.

ISWC'07
The Sixth International Semantic Web Conference
Busan, Korea
November 11-15, 2007
http://iswc2007.semanticweb.org/

ISWC is a major international forum where visionary and state-of-the-art research of all aspects of the Semantic Web are presented. ISWC'07 follows the 1st International Semantic Web Conference (ISWC'02 which was held in Sardinia, Italy, 9-12 June 2002), the 2nd International Semantic Web Conference (ISWC'03 which was held in Florida, USA, 20 - 23 October 2003), 3rd International Semantic Web Conference (ISWC'04 which was held in Hiroshima, Japan, 7 - 11 November 2004), 4th International Semantic Web Conference 2005 (ISWC'05 which was held in Galway, Ireland, 6 - 10 November, 2005) and 5th (ISWC'06 which was held in Athens, GA, USA 5 - 9 November, 2006).

The World-Wide Web continues to grow and new technologies, modes of interactions, and applications are being developed. Building on this growth, Semantic Web technologies aim at providing a shared semantic information space, changing qualitatively our experiences on the Web. As Semantic Web technologies mature and permeate more and more application areas, new research challenges are coming to the fore and some unsolved ones are becoming more acute. These issues include creating and managing Semantic Web content, making Semantic Web applications robust and scalable, organizing and integrating information from different sources for novel uses, making semantics explicit in order to improve our overall experience with information technologies, and thus enabling us to use the wealth of information that is currently available in digital form for addressing our everyday tasks. To foster the exchange of ideas and collaboration, the International Semantic Web Conference brings together researchers in relevant disciplines such as artificial intelligence, databases, social networks, distributed computing, web engineering, information systems, natural language processing, and human-computer interaction.

In addition to this call for papers for the research track, ISWC'07 will include a Semantic Web In Use track, a poster and demonstration track, a doctoral consortium, and a special competition known as the Semantic Web Challenge.

SDM'07
2007 SIAM International Conference on Data Mining
Minneapolis, Minnesota, USA
April 26-28, 2007
http://www.siam.org/meetings/sdm07/

Data mining and knowledge discovery is rapidly becoming an important tool in all walks of human endeavor including science, engineering, industrial processes, healthcare, business, medicine and society. The datasets in these fields are large, complex, and often noisy. Extracting knowledge requires the use of sophisticated, high-performance and principled analysis techniques and algorithms, based on sound statistical foundations. These techniques in turn require powerful visualization technologies; implementations that must be carefully tuned for performance; software systems that are usable by scientists, engineers, and physicians as well as researchers; and infrastructures that support them. For the main conference the program committee seeks outstanding papers in all areas pertaining to data mining and knowledge discovery.

This conference provides a venue for researchers who are addressing these problems to present their work in a peer-reviewed forum. It also provides an ideal setting for graduate students and others new to the field to learn about cutting-edge research by hearing outstanding invited speakers and attending tutorials (included with conference registration). A set of focused workshops are also held on the last day of the conference. The proceedings of the conference are published in archival form, and are also made available on the SIAM web site.

AAAI'07
The Twenty-Second Conference on Artificial Intelligence
Vancouver, British Columbia, Canada
July 22–26, 2007
http://www.aaai.org/Conferences/AAAI/

The Twenty-Second Conference on Artificial Intelligence (AAAI'07) and the collocated Nineteenth Conference on Innovative Applications of Artificial Intelligence (IAAI'07) will be held in Vancouver, British Columbia at the Hyatt Regency Vancouver, from July 22–26.

The AAAI'07 Program Cochairs are Adele Howe, Colorado State University, and Robert C.

Holte at the University of Alberta.

The call for papers for AAAI'07's many programs have now been updated with complete submission information and links to author instructions, paper formatting, and submission sites. AAAI'07 invite your paper submissions to the main technical program, the two special tracks on Artificial Intelligence on the Web (AIW) and Integrated Intelligence (II), the Nectar Program, and the Senior Member Program.

ICTAI'07
The nineteenth International Conference on Tools with Artificial Intelligence
Minneapolis, Minnesota, USA
April 26-28, 2007
http://ictai07.ceid.upatras.gr/

The annual IEEE International Conference on Tools with Artificial Intelligence (ICTAI) provides a major international forum where the creation and exchange of ideas relating to artificial intelligence are fostered among academia, industry, and government agencies. The conference facilitates the cross-fertilization of these ideas and promotes their transfer into practical tools, for developing intelligent systems and pursuing artificial intelligence applications. The ICTAI encompasses all the technical aspects of specifying, developing, and evaluating the theoretical underpinnings and applied mechanisms of AI tools. A selection of the best papers in the conference will be published in a special issue of the International Journal on Artificial Intelligence Tools (IJAIT) (SCI Indexed).

IEEE Computer Society
1730 Massachusetts Ave, NW
Washington, D.C. 20036-1903

Non-profit Org.
U.S. Postage
PAID
Silver Spring, MD
Permit 1398