REQUIREMENTS DOCUMENT

Batch Rename PDF files

By

Jayanth Anantharaman
Apoorva Vinod Gorur
Md Kamruzzaman Sarker

Wright State University,2017

Table of Contents

1 Introduction 2
1.1 Purpose 2
1.2 Scope of the Product 2
1.3 Definitions, acronyms and abbreviations 2
2 General Description 3
2.1 Functional Requirements 3
2.2 Performance Requirement 3
2.3 Maintainability 4
2.4 Platform/Environments 4
3 User Interface 4
4 Acceptance Test 5
5 Appendix 20
6 Test PDFs 21

1. Introduction

This document contains the requirements for Batch Rename PDF files system.

1.1. Purpose

This purpose of this document is to provide documentation of requirements for Batch Rename PDF files system.

1.2. Scope of the Product

The Scope of the product is to only rename the PDF files existing in a folder by parsing the PDF document to extract required details mentioned in one of the below sections.

1.3. Domain specific definitions

Term	Definitions
PDF file	A file with .pdf extension, as per the specification in [2].
Subject Classification	Broad category of the documents to be renamed.
Title	Title of the document. E.g.: “Induction of Decision Trees”
Year	4-digit number representing the calendar year of the published document
Conference/Journal name	Venue of the document published
Author name	Name of the person(s) authored the document
Affiliation	Name of the institution/organization to which authors are affiliated

Table 1 - Definitions of terms

2. General Description

Batch Rename PDF files is a system which will rename the PDF files in a folder in batches with a specific format by extracting relevant information from within the PDF file.

2.1. Functional Requirements:

1. The product must rename the PDF files within a directory by extracting one or many of the below mentioned information from within the PDF file according to the user choice.
    a. The subject classification.
    b. The title of the paper.
    c. The year of publication.
    d. Name of conference or journal of publication
    e. List of author names (First Middle Last)
    f. List of Institutions affiliated with the author.

2. Renaming should follow the below format.
   SUBJECT CLASSIFICATION_TITLE_YEAR_VENUE_AUTHORNAMES_AFFILIATIONS.pdf

3. Product should report the number of files renamed out of the given batch.

4. If any required information mentioned in 1 is missing in the pdf document, rename it with available information from the pdf document.

5. If none of the required information are available, skip renaming the pdf document.

6. If the pdf is password protected or not readable skip renaming the pdf document.

7. Files with .pdf extension should only be considered for renaming.

8. The product should not in any case edit or manipulate the contents of the PDF or delete any existing file

2.2. Performance Requirement:

1. Product should be able to rename a given batch of pdf files with 95% accuracy with reasonable efficiency.
2. Product should not crash or hang.
3. Error handling and show log message to user.

2.3. Maintainability:

Source code should be documented and maintainable.

2.4. Platform/Environments:

Product should be available for Windows, Mac and Linux platforms.

3. User Interface

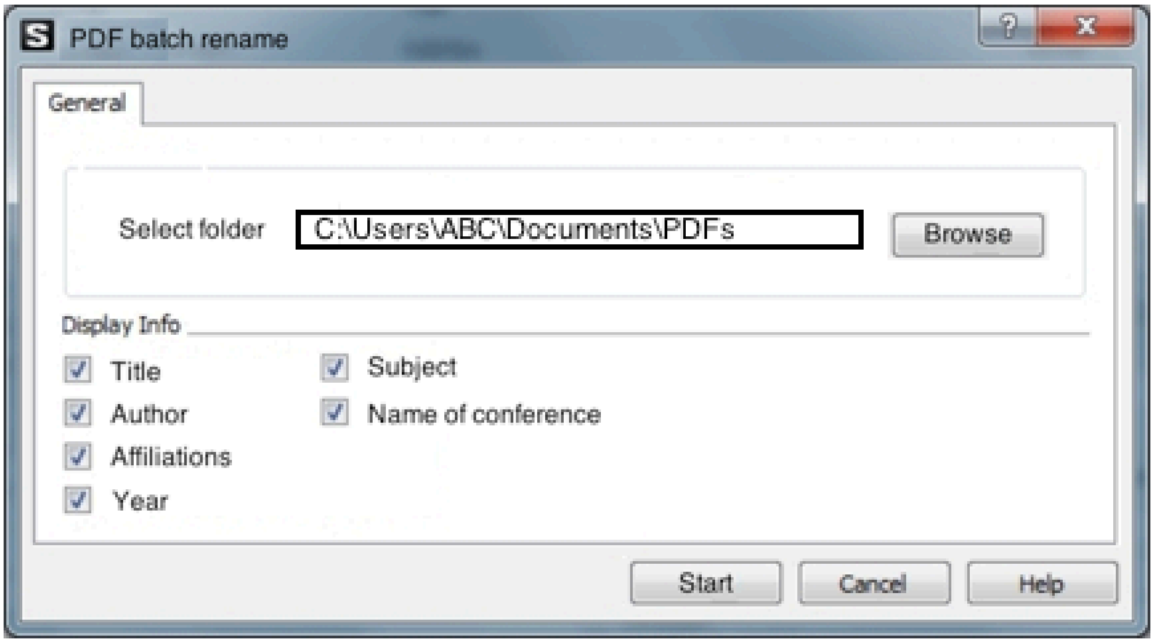


Figure 1: User Interface of PDF Batch Rename files

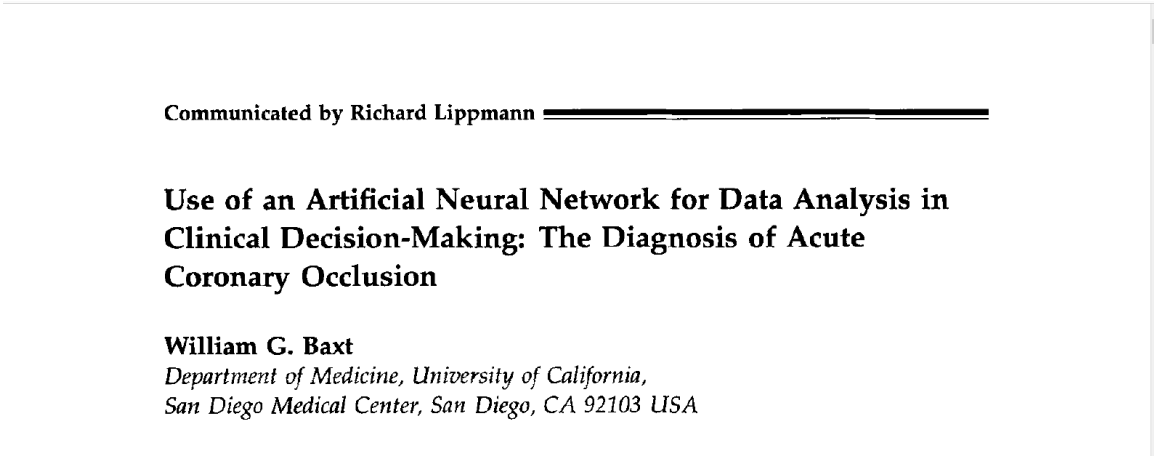
Figure 1 is a sample mock - up of the user interface to be developed in the product. Below are the functionalities expected from the user interface.

1. It must include actionable buttons namely
    a. Browse - Button to input the folder where documents are to be renamed
    b. Checkboxes - Check box indicating the labels to be included while renaming the pdf documents
    c. Start - Initiate the process of renaming the documents

d. Cancel - Gracefully stop the running process
e. Help - To provide user documentation of the product

4. Acceptance Test

Following are some sample snapshots of PDFs from which the product should be able to extract the required information to rename. For simplicity snapshot of the first page of the pdf and extracted information from it is shown first. After the appendix section all pdfs are attached. Complete set of documents are available in the following link [1]



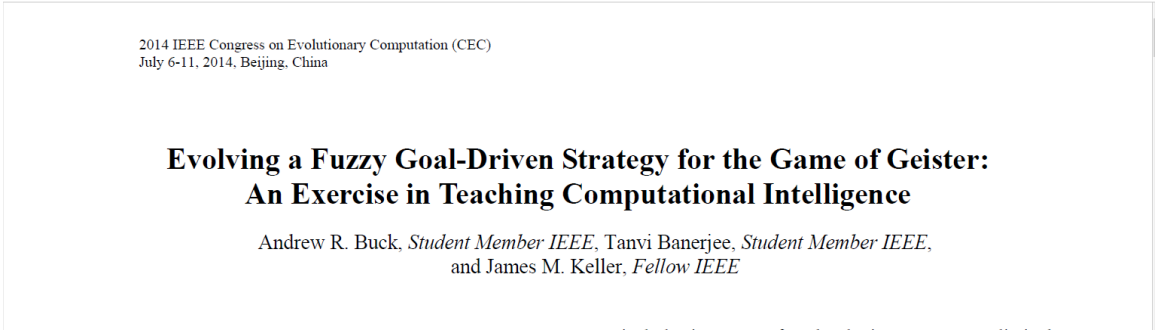
Communicated by Richard Lippmann

Use of an Artificial Neural Network for Data Analysis in Clinical Decision-Making: The Diagnosis of Acute Coronary Occlusion

William G. Baxt
Department of Medicine, University of California, San Diego Medical Center, San Diego, CA 92103 USA

Figure 2: Research Paper Sample 1

Title: Use of an Artificial Neural Network for Data Analysis in Clinical Decision-Making
Author(s): William G. Baxt
Affiliation: University of California
Year: 1990
Subject: Not Available
Name of Conference: Not Available



2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Evolving a Fuzzy Goal-Driven Strategy for the Game of Geister: An Exercise in Teaching Computational Intelligence

Andrew R. Buck, *Student Member IEEE*, Tanvi Banerjee, *Student Member IEEE*, and James M. Keller, *Fellow IEEE*

Figure 3: Research Paper Sample 2

Title: Evolving a Fuzzy Goal-Driven Strategy for the Game of Geister
Author(s):
Andrew R Buck,
Tanvi Banerjee,
James M Keller
Affiliation: IEEE
Year: 2014
Subject: Not Available
Name of Conference: IEEE Congress on Evolutionary Computation

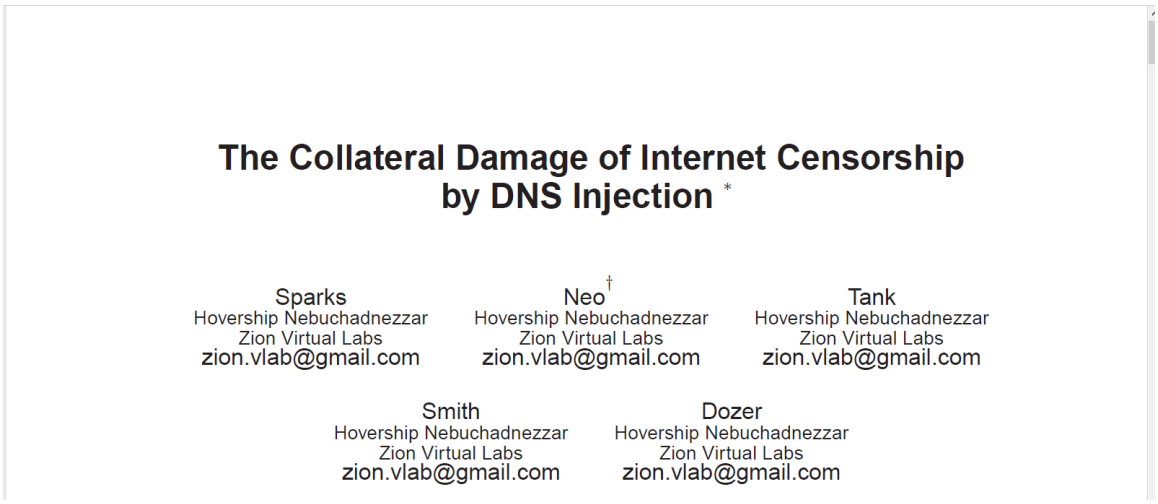
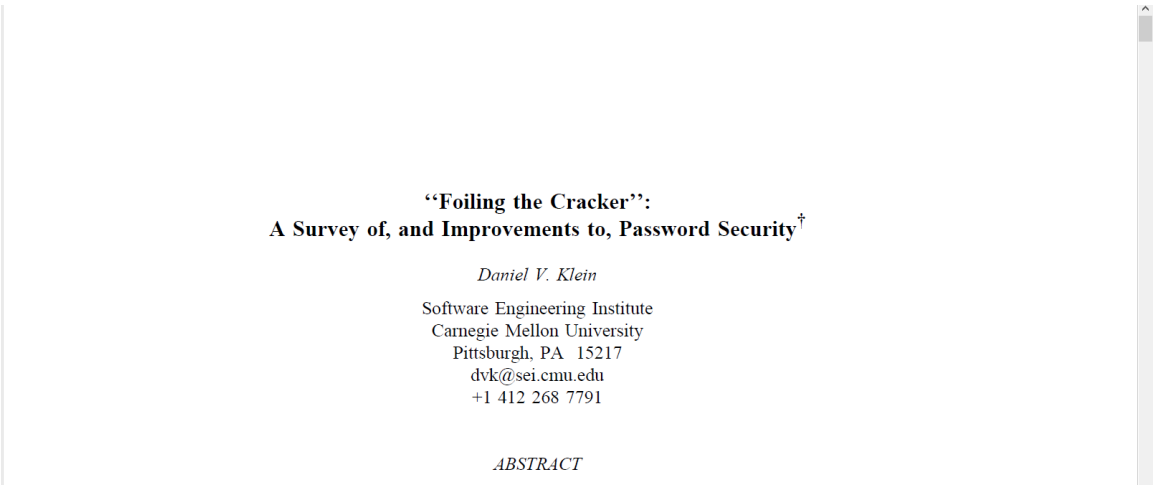


Figure 4: Research Paper Sample 3

Title: The Collateral Damage of Internet Censorship by DNS Injection
Author(s):
Spark,
Neo,
Tank,
Smith, Dozer
Affiliation: Zion Virtual Labs
Year: Not Available
Subject: Not Available
Name of Conference: Not Available

Figure 5: Research Paper Sample 4

Title: Mobile Data Charging: New Attacks and Countermeasures
Author(s):
Chunyi Peng,
Chi-yu Li,
Guan-hua Tu,
Songwu Lu,
Lixia Zhang
Affiliation: University of California
Year: Not Available
Subject: Not Available
Name of Conference: Not Available



''Foiling the Cracker'':
A Survey of, and Improvements to, Password Security[†]

Daniel V. Klein

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15217
dvk@sei.cmu.edu
+1 412 268 7791

ABSTRACT

Figure 6: Research Paper Sample 5

Title: Foiling the Cracker
Author(s): Daniel V Klein
Affiliation: Carnegie Mellon University
Year: Not Available
Subject: Not Available
Name of Conference: Not Available

2016 IEEE International Conference on Systems, Man, and Cybernetics • SMC 2016 | October 9-12, 2016 • Budapest, Hungary

QryGraph: A Graphical Tool for Big Data Analytics

Sanny Schmid, Ilias Gerostathopoulos, Christian Prehofer
Fakultät für Informatik
Technische Universität München
Munich, Germany
{schmidsa, gerostat, prehofer}@in.tum.de

Figure 7: Research Paper Sample 6

Title: QryGraph: A Graphical Tool for Big Data Analytics
Author(s):
Sanny Schmid,
Ilias Gerostathopoulos,
Christian Prehofer
Affiliation: Fakultät für Informatik Technische Universität München Munich, Germany
Year: Not Available
Subject: Not Available
Name of Conference: Not Available

2016 IEEE TrustCom-BigDataSE-ISPA

Rethinking High Performance Computing System Architecture for Scientific Big Data Applications

Yong Chen*, Chao Chen*, Yanlong Yin†, Xian-He Sun†, Rajeev Thakur‡, William D Gropp§

*Department of Computer Science, Texas Tech University, Email: yong.chen@ttu.edu, chao.chen@ttu.edu
†Department of Computer Science, Illinois Institute of Technology, Email: yyin2@ttu.edu, sun@ttu.edu
‡Mathematics and Computer Science Division, Argonne National Laboratory, Email: thakur@mcs.anl.gov
§Department of Computer Science, University of Illinois Urbana-Champaign, Email: wgropp@illinois.edu

Figure 8: Research Paper Sample 7

Title: Rethinking High Performance Computing System Architecture for Scientific Big Data Applications
Author(s):
Yong Chen[1],
Chao Chen[2],
Xian-He Sun Rajeev Thakur[3],
William D Gropp[4]
Affiliation:
Texas Tech University[1],
Illinois Institute of technology[2],
Argonne National Laboratory[3],
University of Illinois Urbana-Champaign[4]
Year: 2016
Subject: Not Available
Name of Conference:IEEE TrustCom-BigDataSE-ISPA

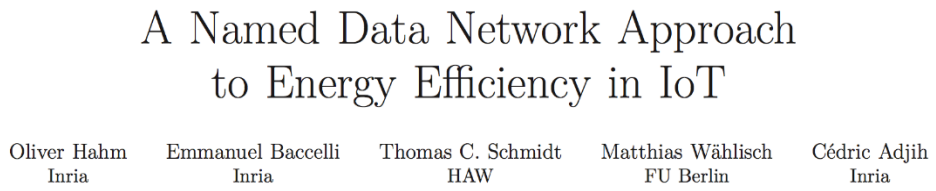


Figure 9: Research Paper Sample 8

Title: A Named Data Network Approach to Energy Efficiency in IoT
Author(s):
Oliver Hahm Inria,
Thomas C. Schmidt HAW,
Matthias Wählisch FU Berlin,
Cédric Adjih Inria
Affiliation: Not Available
Year: 2016
Subject: Not Available
Name of Conference: Not Available

IVYSPRING INTERNATIONAL PUBLISHER

International Journal of Medical Sciences

2017; 14(2): 181-190. doi: 10.7150/ijms.17119

Research Paper

Immune Cell Repertoire and Their Mediators in Patients with Acute Myocardial Infarction or Stable Angina Pectoris

Wenwen Yan[1], Yanli Song[2], Lin Zhou[1], Jinfa Jiang[1], Fang Yang[3], Qianglin Duan[1], Lin Che[1], Yuqin Shen[1]✉, Haoming Song[1]✉, Lemin Wang[1]✉

1. Department of Cardiology, Tongji Hospital, Tongji University School of Medicine, Shanghai 200065, China;
2. Department of Emergency Medicine, Tongji Hospital, Tongji University School of Medicine, Shanghai 200065, China;
3. Department of Experimental Diagnosis, Tongji Hospital, Tongji University School of Medicine, Shanghai 200065, China.

✉ Corresponding authors: Lemin Wang, Haoming Song, Yuqin Shen, Department of Cardiology, Tongji Hospital, Tongji University School of Medicine, 389 Xincun Rd, Putuo District, Shanghai 200065, China; Tel: 86 21 66111329, Fax: 86 21 66111329, E-mail: wanglemin@tongji.edu.cn; songhao-ming@163.com; sy-1963@126.com.

Received: 2016.08.05; Accepted: 2016.12.21; Published: 2017.02.08

Figure 10: Research Paper Sample 9

Title: Immune Cell Repertoire and Their Mediators in Patients with Acute Myocardial Infarction or Stable Angina Pectoris

Author(s):

Wenwen Yan,
Yanli Song,
Lin Zhou,
Jinfa Jiang,
Fang Yang,
Qianglin Duan,
Lin Che,
Yuqin Shen,
Haoming Song,
Lemin Wang

Affiliation: Tongji University School of Medicine, Shanghai

Year: 2017

Subject: Not Available

Name of Conference: International Journal of Medical Sciences

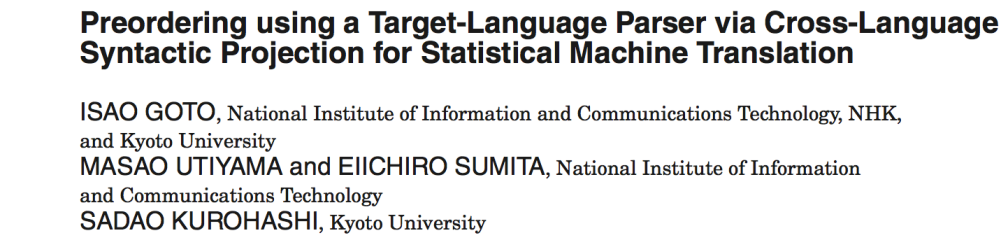


Figure 11: Research Paper Sample 10

Title: Preordering using a Target-Language Parser via Cross-Language Syntactic Projection for Statistical Machine Translation
Author(s):
ISAO GOTO[1],
MASAO UTIYAMA[1],
EIICHIRO SUMITA[1],
SADAO KUROHASHI[2]
Affiliation:
National Institute of Information and Communications Technology[1],
Kyoto University[2]
Year: Not Available
Subject: Not Available
Name of Conference: Not Available

Available online at www.sciencedirect.com
ScienceDirect
Decision Support Systems 42 (2006) 674–689
Decision Support Systems
www.elsevier.com/locate/dsw

A new approach to classification based on association rule mining

Guoqing Chen *, Hongyan Liu, Lan Yu, Qiang Wei, Xing Zhang

Department of Management Science and Engineering, School of Economics and Management, Tsinghua University, Beijing 100084, China

Received 19 February 2004; received in revised form 9 March 2005; accepted 9 March 2005
Available online 25 July 2005

Figure 12: Research Paper Sample 11

Subject Classification: Not available
Title: A new approach to classification based on association rule mining
Year: received-2004, accepted-2005
Conference/Journal name: Not available
Author name:
Guoqing Chen[1],
Hongyan Liu[1],
Lan Yu[1],
Qiang Wei[1],
Xing Zhang[1]
Affiliation: Department of Management Science and Engineering, School of Economics and Management, Tsinghua University, Beijing 100084, China[1]

Machine Learning 1: 81–106, 1986

Induction of Decision Trees

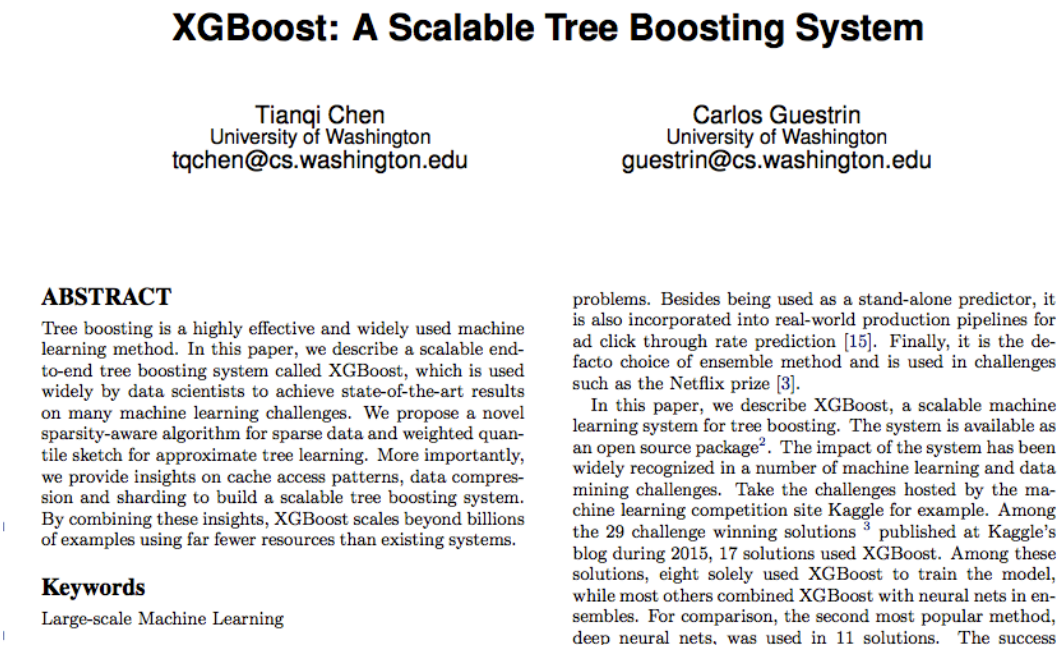
J.R. QUINLAN (munnari!nswitgould.oz!quinlan@seismo.css.gov)
Centre for Advanced Computing Sciences, New South Wales Institute of Technology, Sydney 2007, Australia

(Received August 1, 1985)

Key words: classification, induction, decision trees, information theory, knowledge acquisition, expert systems

Figure 13: Research Paper Sample 12

Subject Classification: classification, induction, decision trees, information theory, knowledge acquisition, expert systems
Title: Induction of Decision Trees
Year:1986
Conference/Journal name: Kluwer Academic Publishers, Boston
Author name:
J.R. QUINLAN
Affiliation: Centre for Advanced Computing Sciences, New South Wales Institute of Technology, Sydney, Australia



XGBoost: A Scalable Tree Boosting System

Tianqi Chen
University of Washington
tqchen@cs.washington.edu

Carlos Guestrin
University of Washington
guestrin@cs.washington.edu

ABSTRACT

Tree boosting is a highly effective and widely used machine learning method. In this paper, we describe a scalable end-to-end tree boosting system called XGBoost, which is used widely by data scientists to achieve state-of-the-art results on many machine learning challenges. We propose a novel sparsity-aware algorithm for sparse data and weighted quantile sketch for approximate tree learning. More importantly, we provide insights on cache access patterns, data compression and sharding to build a scalable tree boosting system. By combining these insights, XGBoost scales beyond billions of examples using far fewer resources than existing systems.

Keywords

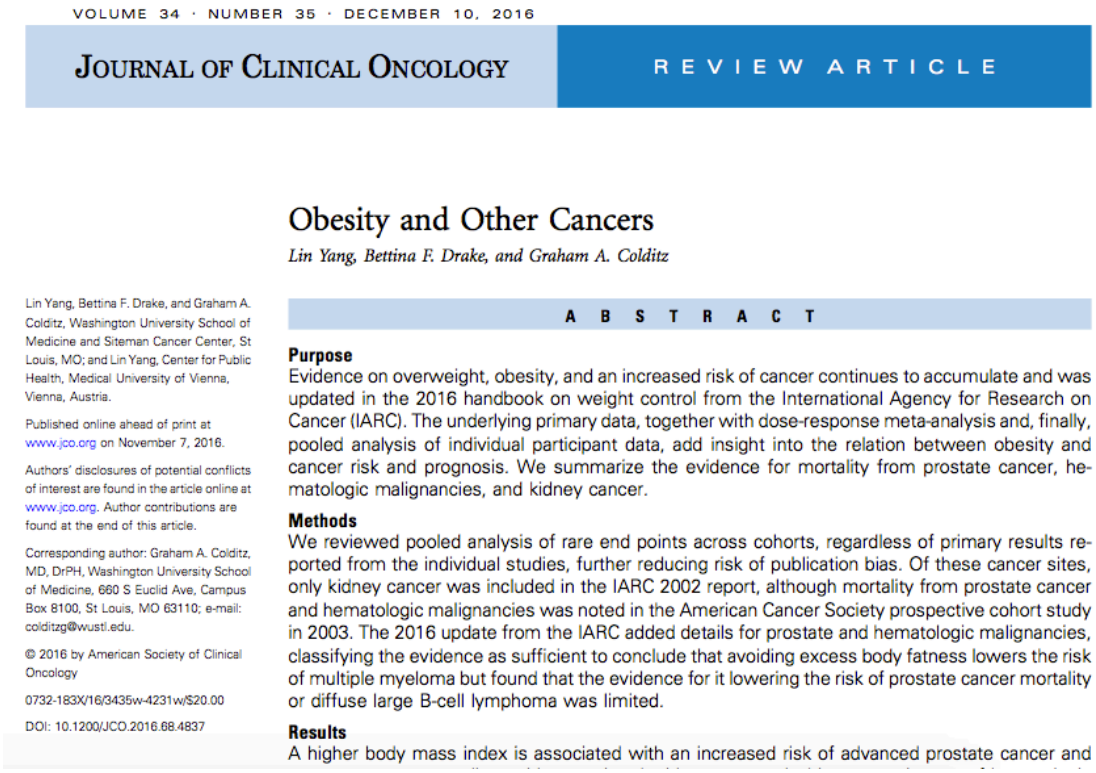
Large-scale Machine Learning

problems. Besides being used as a stand-alone predictor, it is also incorporated into real-world production pipelines for ad click through rate prediction [15]. Finally, it is the de-facto choice of ensemble method and is used in challenges such as the Netflix prize [3].

In this paper, we describe XGBoost, a scalable machine learning system for tree boosting. The system is available as an open source package[2]. The impact of the system has been widely recognized in a number of machine learning and data mining challenges. Take the challenges hosted by the machine learning competition site Kaggle for example. Among the 29 challenge winning solutions [3] published at Kaggle's blog during 2015, 17 solutions used XGBoost. Among these solutions, eight solely used XGBoost to train the model, while most others combined XGBoost with neural nets in ensembles. For comparison, the second most popular method, deep neural nets, was used in 11 solutions. The success

Figure 14: Research Paper Sample 13

Subject Classification: Large-scale Machine Learning
Title: XGBoost: A Scalable Tree Boosting System
Year: Not Available
Conference/Journal name: Not Available
Author name:
Carlos Guestrin[1],
Carlos Guestrin[1]
Affiliation: University of Washington[1]



VOLUME 34 · NUMBER 35 · DECEMBER 10, 2016

JOURNAL OF CLINICAL ONCOLOGY

REVIEW ARTICLE

Obesity and Other Cancers

Lin Yang, Bettina F. Drake, and Graham A. Colditz

Lin Yang, Bettina F. Drake, and Graham A. Colditz, Washington University School of Medicine and Siteman Cancer Center, St Louis, MO; and Lin Yang, Center for Public Health, Medical University of Vienna, Vienna, Austria.

Published online ahead of print at www.jco.org on November 7, 2016.

Authors' disclosures of potential conflicts of interest are found in the article online at www.jco.org. Author contributions are found at the end of this article.

Corresponding author: Graham A. Colditz, MD, DrPH, Washington University School of Medicine, 660 S Euclid Ave, Campus Box 8100, St Louis, MO 63110; e-mail: colditzg@wustl.edu.

0732-183X/16/3435w-4231w/$20.00

DOI: 10.1200/JCO.2016.68.4837

A B S T R A C T

Purpose

Evidence on overweight, obesity, and an increased risk of cancer continues to accumulate and was updated in the 2016 handbook on weight control from the International Agency for Research on Cancer (IARC). The underlying primary data, together with dose-response meta-analysis and, finally, pooled analysis of individual participant data, add insight into the relation between obesity and cancer risk and prognosis. We summarize the evidence for mortality from prostate cancer, hematologic malignancies, and kidney cancer.

Methods

We reviewed pooled analysis of rare end points across cohorts, regardless of primary results reported from the individual studies, further reducing risk of publication bias. Of these cancer sites, only kidney cancer was included in the IARC 2002 report, although mortality from prostate cancer and hematologic malignancies was noted in the American Cancer Society prospective cohort study in 2003. The 2016 update from the IARC added details for prostate and hematologic malignancies, classifying the evidence as sufficient to conclude that avoiding excess body fatness lowers the risk of multiple myeloma but found that the evidence for it lowering the risk of prostate cancer mortality or diffuse large B-cell lymphoma was limited.

Results

A higher body mass index is associated with an increased risk of advanced prostate cancer and

Figure 15: Research Paper Sample 14

Subject Classification: Not Available
Title: Obesity and Other Cancers
Year: 2016
Conference/Journal name: JOURNAL OF CLINICAL ONCOLOGY
Author name:

Lin Yang[1,2],
Bettina F. Drake[1], and
Graham A. Colditz[1]

Affiliation:

Washington University School of Medicine and Siteman Cancer Center, St Louis, MO[1].

Center for Public Health, Medical University of Vienna, Vienna, Austria[2].

RESEARCH ARTICLE

MATERIALS SCIENCE

Experimental test of Landauer's principle in single-bit operations on nanomagnetic memory bits

10.1126/sciadv.1501492

Jeongmin Hong,[1] Brian Lambson,[2] Scott Dhuey,[3] Jeffrey Bokor[1]*

Minimizing energy dissipation has emerged as the key challenge in continuing to scale the performance of digital computers. The question of whether there exists a fundamental lower limit to the energy required for digital operations is therefore of great interest. A well-known theoretical result put forward by Landauer states that any irreversible single-bit operation on a physical memory element in contact with a heat bath at a temperature T requires at least $k_B T \ln(2)$ of heat be dissipated from the memory into the environment, where k_B is the Boltzmann constant. We report an experimental investigation of the intrinsic energy loss of an adiabatic single-bit reset operation using nanoscale magnetic memory bits, by far the most ubiquitous digital storage technology in use today. Through sensitive, high-precision magnetometry measurements, we observed that the amount of dissipated energy in this process is consistent (within 2 SDs of experimental uncertainty) with the Landauer limit. This result reinforces the connection between "information thermodynamics" and physical systems and also provides a foundation for the development of practical information processing technologies that approach the fundamental limit of energy dissipation. The significance of the result includes insightful direction for future development of information technology.

Figure 16: Research Paper Sample 15

Subject Classification: Not Available
Title: Experimental test of Landauer's principle in single-bit operations on nanomagnetic memory bits
Year: Not Available
Conference/Journal name: Not Available
Author names:

Jeongmin Hong[1],
Brian Lambson[2],
Scott Dhuey[3],
Jeffrey Bokor[1]
Affiliation:
Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA 94720, USA[1].
Haynes and Boone LLP, 525 University Avenue, Palo Alto, CA 94301, USA[2].
The Molecular Foundry, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA[3].

Collected 15 test pdfs is attached after the appendix.

Appendix

[1] https://github.com/md-k-sarker/PDF-Renamer/tree/master/TestDocument
[2] https://en.wikipedia.org/wiki/Portable_Document_Format
[3] https://github.com/md-k-sarker/PDF-Renamer/blob/master/TestDocument/Sarker/10.1.1.167.3624.pdf

Communicated by Richard Lippmann

Use of an Artificial Neural Network for Data Analysis in Clinical Decision-Making: The Diagnosis of Acute Coronary Occlusion

William G. Baxt
Department of Medicine, University of California, San Diego Medical Center, San Diego, CA 92103 USA

A nonlinear artificial neural network trained by backpropagation was applied to the diagnosis of acute myocardial infarction (coronary occlusion) in patients presenting to the emergency department with acute anterior chest pain. Three-hundred and fifty-six patients were retrospectively studied, of which 236 did not have acute myocardial infarction and 120 did have infarction. The network was trained on a randomly chosen set of half of the patients who had not sustained acute myocardial infarction and half of the patients who had sustained infarction. It was then tested on a set consisting of the remaining patients to which it had not been exposed. The network correctly identified 92% of the patients with acute myocardial infarction and 96% of the patients without infarction. When all patients with the electrocardiographic evidence of infarction were removed from the cohort, the network correctly identified 80% of the patients with infarction. This is substantially better than the performance reported for either physicians or any other analytical approach.

1 Introduction

Decision-making under uncertainty is often fraught with great difficulty when the data on which the decision is based are imprecise and poorly linked to predicted outcome (Holloway 1979). Clinical diagnosis is an example of such a setting (Moskowitz et al. 1988) because multiple, often unrelated, disease states can present with similar or identical historical, symptomalogic, and clinical data. In addition, singular disease states do not always present with the same historical, symptomalogic, and clinical data. As a result, physician accuracy in diagnosing many of these diseases is often disappointing. A number of approaches have been developed to analyze data collected during patient evaluation to improve on diagnostic accuracy, but none of these approaches has been able to improve significantly on the performance of well-trained physicians (Reggia and Tuhrim 1985; Szolovits et al. 1988). The question still remains as to

whether there is any means by which the data available in the clinical setting can be analyzed to yield information that can be utilized to improve diagnostic accuracy.

Acute myocardial infarction is an example of a disease process that has been difficult to diagnose accurately. A considerable number of methodologies have been developed in attempts to improve on the diagnostic accuracy of physicians in identifying the presence of acute myocardial infarction (Pozen et al. 1977, 1980, 1984; Goldman et al. 1982, 1988a; Patrick et al. 1976, 1977; Lee et al. 1985, 1987a,b; Tierney et al. 1985). Stepwise discriminate analysis (Pozen et al. 1977), logistic regression (Pozen et al. 1980), recursive partition analysis (Goldman et al. 1982), and pattern recognition (Patrick et al. 1976, 1977) have been utilized. The best of these approaches has performed with the same detection rate (sensitivity) (88%) and slightly better false alarm rate (1.0-specificity) (26% vs. 29%) than physicians (Goldman et al. 1988a). The following reports on the use of artificial neural network techniques (Widrow and Hoff 1960; Rumelhart et al. 1986; McClelland and Rumelhart 1988; Weigend et al. 1990; Mulsant and Servan-Schreiber 1988; Hudson et al. 1988; Smith et al. 1988; De Roach 1989; Saito and Nakano 1988; Marconi et al. 1989) to determine if the data collected during the routine evaluation of patients for acute myocardial infarction contain previously inapparent information that can be used to improve on the diagnostic accuracy of predicting the presence of acute myocardial infarction.

2 Methods

The nonlinear artificial neural network was a multilayer perceptron trained with backpropagation by use of the McClelland and Rumelhart simulator (McClelland and Rumelhart 1988). Figure 1 depicts the topology of the network utilized.

The network was trained by dividing the available data into a training set and a test set. Training took place by choosing input patterns from the training set and allowing activation to flow from the inputs through the hidden units to the output unit. The value of the output unit activation was then compared to the documented diagnosis for each pattern. The difference (error) between the actual activation of the output unit and the correct value was then utilized by the backpropagation algorithm (Rumelhart et al. 1986; McClelland and Rumelhart 1988) to modify all weights of the network so that future outputs approximate the correct diagnosis.

Because most patients presenting to the emergency department with anterior chest pain are not suffering from acute myocardial infarction (Goldman et al. 1988a), a subset of patients with a much greater probability of having sustained infarction were chosen for this study. To this end, only patients admitted to the coronary care unit were studied. In this

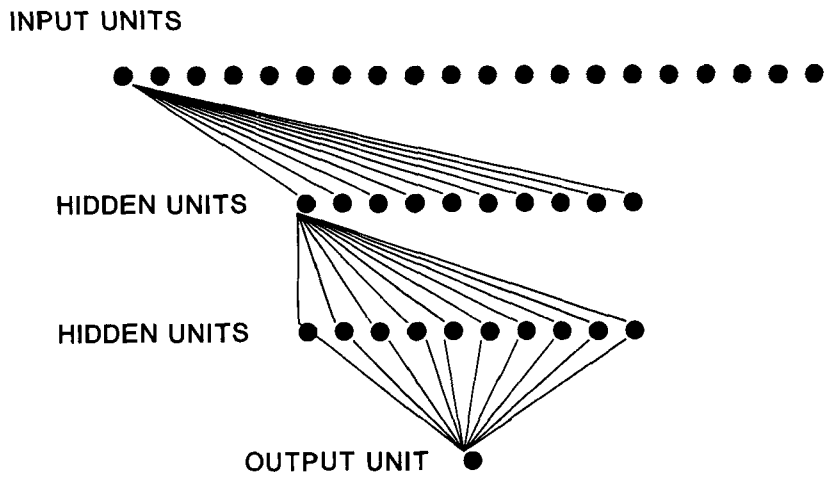


Figure 1: $20 \times 10 \times 10 \times 1$ nonlinear artificial neural network. The network has 20 input units, two layers of 10 hidden units each, and one output unit. Only the connections from one input and one hidden unit from each layer are shown. The network simulator program was run on a 80386 microcomputer with an 80387 math coprocessor running at 20 mHz. Epsilon was set at 0.05. Alpha was set at 0.9. Initial weights were random. Training times ranged between 8 and 48 hr.

way, the network was presented with the potentially most challenging pattern sets to differentiate. A retrospective chart review was performed on 356 patients who were admitted through the emergency department to the coronary care unit to rule out the presence of infarction. Forty-one variables reported to be predictive of the presence of acute myocardial infarction (Pozen et al. 1977, 1980, 1984; Goldman et al. 1982, 1988a; Patrick et al. 1976, 1977; Lee et al. 1985, 1987a,b,; Tierney et al. 1985) (depicted in Table 1) were collected on all patients from the emergency department record. The manner in which the presence or absence of infarction was determined was also documented from the inpatient record. The presence of infarction was confirmed as reported elsewhere (Goldman et al. 1988a).

The input patterns were generated by a specially written program that coded most of the clinical input variables in a binary manner such that 1 equalled the presence of a finding and 0 the absence of a finding. *Patient age*, blood pressure, pulse, and pain intensity were coded as analog values between 0.0 and 1.0. The target value for the output was coded as 0 for the subsequently confirmed absence of acute myocardial infarction and 1 for the confirmed presence of infarction.

History	Past History	Examination	Electrocardiogram findings
Age*	Past AMI*	Systolic BP	2 mm ST elevation*
Sex*	Angina*	Diastolic BP	1 mm ST elevation*
Location of pain*	Congestive heart failure	Pulse	ST depression*
Intensity of pain	Diabetes*	Jugular venous distension*	T wave inversion*
Duration of pain	Hypertension*	Rales*	Peaked T wave
Radiation of pain	Family history AMI	Third heart sound	Premature ventricular contractions
Pain pleuritic	High cholesterol	Fourth heart sound	Heart block
Similar to past AMI	Coffee	Edema	Intraventricular conduction defect
Response to pressure	Cigarettes		Significant ischemic change*
Response to nitroglycerin*			
Nausea and vomiting*			
Diaphoresis*			
Syncope*			
Shortness of breath*			
Palpitations*			

Table 1: Input Variables. Variables marked with "*" utilized in final pattern sets.

To find a predictive set of input variables, different input pattern formats utilizing different numbers and combinations of the input variables were tested on networks that had as little as 5 to as many as 41 input units. To find a more optimal network architecture, different numbers of hidden units arranged in different numbers of layers were tested. A network with 20 inputs and 2 layers of 10 hidden units each, as depicted in Figure 1, utilizing the 20 clinical input variables noted in Table 1, was chosen on the basis of this analysis. Learning was followed by totaling the sum square (TSS) error over the pattern set (Rumelhart et al. 1986). Input patterns were presented to the network and learning epochs run

until the TSS ceased decreasing. The final weights derived from a training session were then saved for use in testing.

Testing of a network was accomplished by using the weights derived in the training set and presenting the network with patterns to which it had *not* been exposed. Performance was scored as correct if the activation of the output unit was equal to or greater than 0.8 when the target was 1 or when the activation of the output unit was equal to or less than 0.2 when the target was 0. The output unit activation in this study was always between 0 and 0.2 and 0.8 and 1.0. Detection rate (sensitivity) was defined as the number of patients in a test population correctly diagnosed as having a disease divided by the total number in the test set with the disease. False alarm rate (specificity) was defined as the number of patients in a test population correctly diagnosed as not having a disease divided by the total number in the test set without the disease.

3 Results

The network was trained utilizing a randomly chosen subset of patterns derived from the initial group of 356 patients. Half of the patients who had not sustained acute myocardial infarctions and half of the patients who had sustained infarctions were selected. The subset consisted of 118 patients who were diagnosed as not having sustained an infarction and 60 patients who were diagnosed as having sustained an infarction. The final TSS reached was 0.044 error per pattern. The network was then tested on the remaining 178 patients (118 noninfarction, 60 infarction) to which it had not been exposed. The network correctly diagnosed 55 of the 60 patients with infarction and 113 of the 118 patients without infarction.

This process was repeated utilizing the second pattern set for network training and the first set for testing. The initial TSS achieved during training was 0.02 error per pattern. The network correctly diagnosed 56 of the 60 patients with acute myocardial infarction and 113 of the 118 patients without infarction on the test set. The summed results of the two test sets are depicted in Table 2. The network performed with a detection rate of 92% and a false alarm rate of 96%.

	Noninfarction	Infarction
Correct	226	111
Incorrect	−10	−9

Table 2: All Patients. Detection rate (sensitivity), 92%; false alarm rate (1.0-specificity), 4%.

	Noninfarction	Infarction
Correct	47	44
Incorrect	−4	−7

Table 3: Infarction Patients without ST Elevation. Detection rate (sensitivity), 86%; false alarm rate (1.0-specificity), 8%.

A significant number of patients who present to the emergency department who have sustained acute myocardial infarction have clear-cut electrocardiographic evidence of infarction. The real diagnostic challenge arises in those patients who have sustained infarction, but do not have clear-cut evidence of infarction on their initial electrocardiogram. When such patients were omitted from one study that attempted to improve on physician diagnostic performance, detection fell significantly (Goldman et al. 1988b).

To determine if this approach could effectively identify new information under the most challenging circumstances, the network was further trained and tested on those patients without clear-cut electrocardiographic evidence of acute myocardial infarction. Fifty-two percent of the patients with a documented infarction had acute ST segment elevation on their initial electrocardiogram and none of the patients without infarction had this finding. To study the effect of eliminating such patients, the network was trained on a pattern set derived from half of the patients who sustained infarctions who did not have ST elevation on their initial electrocardiogram along with an equal number of randomly selected patients who had not sustained infarctions. The network was then tested on the second half of the patients who had sustained infarctions who did not have ST elevation on their initial electrocardiogram along with a randomly chosen equal number of patients from the group that had not sustained infarctions. As above, the process was then reversed utilizing the second pattern set for training and the first set for testing. The results are summarized in Table 3. The network performed with a detection rate of 86% and a false alarm rate of 92%, indicating that network performance was not dependent on the presence of ST elevation on the initial electrocardiogram.

Eighty-three percent of the patients with a documented acute myocardial infarction had either acute ST elevation or new ischemic change on their initial electrocardiogram and none of the patients without infarction had this finding. To further study the effect of clear-cut electrocardiographic markers, a set of patients whose initial electrocardiogram showed neither ST elevation nor new ischemic change were identified.

There were 20 such patients. These patients were combined with 20 randomly chosen patients who had not sustained infarction. Because of the small sample size, a leave-one-out strategy was used to test network performance. Input patterns for training were derived from 19 of the 20 patients who had sustained infarction who had neither ST elevation nor significant ischemic change on their initial electrocardiogram along with 20 randomly chosen patients who had not sustained infarction. Twenty such sets of training data were constructed by removing a different infarction patient in each set. The network was trained on each of these pattern sets and tested on the one infarction patient that had been removed. The network correctly identified 16 of the 20 patients with infarction (detection rate 80%), further indicating that network performance was not predominantly dependent on electrocardiographic markers of infarction.

4 Discussion

These data reveal that the artificial neural network had a detection rate of 92% and a false alarm rate of 4%, whereas the best previously reported performance had a detection rate of 88% and a false alarm rate of 26%.

Although these results are encouraging, future studies will need to address some of the questions that were not fully answered in this study. The proof that the nonlinear artificial neural network identified and utilized new information rests on the improvement of diagnostic accuracy derived from comparisons to studies reported in the literature. The results reported here are, thus, compared to studies based on a different set of data. Valid comparisons between methodologies must use the same data sets. In addition, the physician performance on the data set studied herein was not determined and may have been better than that described in the literature. Absolute conclusions about comparative performance will need to be derived from the prospective study of this question. Further, the studies to which these data were compared evaluated all patients presenting to the emergency department with nontraumatic chest pain. This study analyzed data collected only from patients who were admitted to the coronary care unit. The consequence of this will require study.

The good performance afforded by the network deserves comment. Previously utilized statistical strategies have been based on one of three approaches: (1) tree structure rule-based interrelationships, (2) linear pattern matching, or (3) statistical probability calculations (Szolovits et al. 1988). All of these methods are heavily dependent on the consistency of input data for proper performance. One of the striking aspects about the presentation of most disease states is the lack of consistency in their presentation. This emanates from both vague and imprecise clinical histories as well as marked variations in the symptom clusters

and clinical findings with which identical disease processes can present. Decision modalities that are highly dependent on consistency of input to arrive at correct diagnostic closure will perform poorly in this setting. All of the other approaches to clinical decision-making alluded to above are based on a highly structured set of rules or statistical probability prediction that are dependent on the accuracy of input data. One possible reason for the good performance of the artificial neural network is that the nonlinear statistical analysis of the data performed tolerates a considerable amount of imprecise and incomplete input data. These networks appear to be able to cope with the subtle variations in the way disease processes present without making categorical decisions solely driven by these variations. The networks appear to be able to discover implicit higher order conditional dependencies in patterns that are not apparent on face value and to utilize these dependencies to derive generalized rules that are resistant to most minor input perturbations. Specifically, the network can shift from one set of input variables to another and still make accurate prediction based on the actual data at hand. It is this resistance to the perversion of accurate generalizations that enables the network to function more accurately in the clinical environment.

The network used input data that are routinely available to and utilized by physicians screening patients for the presence of acute myocardial infarction. The network simply discovered relationships in these data that are evidently not immediately apparent to physicians and was able to use these to come to a more accurate diagnostic closure. Because these relationships can be made explicit by studying the network weighting of input data, physicians could potentially utilize this information to make a more accurate diagnosis. The actual possibility of this will depend on the complexity of the interrelationships defined by the network. It has been demonstrated that when networks with more than one hidden layer are required to achieve optimal training, the solutions are often distributed over multiple units and are difficult to identify (Weigend et al. 1990). However, if these relationships can be elucidated, the use of the network may be unnecessary.

These observations must be validated by extending this study to a larger number of patients, followed by the prospective testing of the relationships identified by the network. If these results hold up to such scrutiny, the improvement in predictive accuracy could have a substantial impact on the reduction of health care costs. Furthermore, these techniques may be able to be extended to other clinical settings.

Acknowledgments

I thank Dr. David Zipser for his help with this study and Kathleen James for her help in the preparation of this manuscript.

References

De Roach, J. N. 1989. Neural networks — An artificial intelligence approach to the analysis of clinical data. *Austral. Phys. Engineer. Sci. Med.* **12**, 100–106.

Goldman, L., Weinberg, M., Weisberg, M., Olshen, R., Cook, E. F., Sargent, R. K., Lamas, G. A., Dennis, C., Wilson, C., Deckelbaum, L., Fineberg, H., Stiratelli, R., and the Medical House Staffs at Yale–New Haven Hospital and Brigham and Women's Hospital. 1982. A computer-derived protocol to aid in the diagnosis of emergency room patients with acute chest pain. *N. Engl. J. Med.* **307**, 588–596.

Goldman, L., Cook, E. F., Brand, D. A., Lee, T. H., Rouan, G. W., Weisberg, M. C., Acampora, D., Stasiulewicz, C., Walshon, J., Terranova, G., Gottlieb, L., Kobernick, M., Goldstein-Wayne, B., Copen, D., Daley, K., Brandt, A. A., Jones, D., Mellors, J., and Jakubowski, R. 1988a. A computer protocol to predict myocardial infarction in emergency department patients with chest pain. *N. Engl. J. Med.* **318**, 797–803.

Goldman, L., Cook, E. F., Brand, D. A., Lee, T. H., and Rouan, G. W. 1988b. Letter to the editor. *N. Engl. J. Med.* **319**, 792.

Holloway, C. A. 1979. Behavioral assumptions and limitations of decision analysis. In *Decision Making Under Uncertainty: Models and Choices*, C. A. Holloway, ed., pp. 436–455. Prentice-Hall, Englewood Cliffs, NJ.

Hudson, D. L., Cohen, M. E., Anderson, M. F. 1988. Determination of testing efficacy in carcinoma of the lung using a neural network model. *Symp. Comput. Applic. Med. Care Proc.* **12**, 251–255.

Lee, T. H., Cook, E. F., Weisberg, M., Sargent, R. K., Wilson, C., and Goldman, L. 1985. Acute chest pain in the emergency ward: Identification and examination of low-risk patients. *Arch. Intern. Med.* **145**, 65–69.

Lee, T. H., Rouan, G. W., Weisberg, M. C., Brand, D. A., Cook, F., Acampora, D., Goldman, L., and the Chest Pain Study Group; Boston, MA; New Haven, Danbury, and Milford, CT; and Cincinnati, OH. 1987a. Sensitivity of routine clinical criteria for diagnosing myocardial infarction within 24 hours of hospitalization. *Ann. Intern. Med.* **106**, 181–186.

Lee, T. H., Rouan, G. W., Weisberg, M. C., Brand, D. A., Acampora, D., Stasiulewicz, C., Walshon, J., Terranova, G., Gottlieb, L., Goldstein-Wayne, B., Copen, D., Daley, K., Brandt, A. A., Mellors J., Jakubowski, R., Cook, E. F., and Goldman, L. 1987b. Clinical characteristics and natural history of patients with acute myocardial infarction sent home from the emergency room. *Am. J. Cardiol.* **60**, 219–224.

Marconi, L., Scalia, F., Ridella, S., Arrigo, P., Mansi, C., and Mela, G. S. 1989. An application of back propagation to medical diagnosis. *Symp. Comput. Applic. Med. Care Proc.*, in press.

McClelland, J. L., and Rumelhart, D. E., eds. 1988. Training hidden units. In *Explorations in Parallel Distributed Processing*, pp. 121–160. MIT Press, Cambridge, MA.

Moskowitz, A. J., Kuipers, B. J., and Kassirer, J. P. 1988. Dealing with uncertainty, risks, and trade-offs in clinical decisions. A cognitive science approach. *Ann. Intern. Med.* **108**, 435–449.

Mulsant, G. H., and Servan-Schreiber, E. 1988. A connectionist approach to the diagnosis of dementia. *Symp. Comput. Applic. Med. Care Proc.* **12**, 245–250.

Patrick, E. A., Margolin, G., Sanghvi, V., and Uthurusamy, R. 1976. Pattern recognition applied to early diagnosis of heart attacks. In *Proceedings of the IEEE 1976 Systems, Man, and Cybernetics Conference*, Washington, D.C., November 1–3, pp. 403–406.

Patrick, E. A., Margolin, G., Sanghvi, V., and Uthurusamy, R. 1977. Pattern recognition applied to early diagnosis of heart attacks. In *Proceedings of the 1977 International Medical Information Processing Conference (MEDINFO)*, Toronto, August 9–12, pp. 203–207.

Pozen, M. W., Stechmiller, J. K., and Voigt, G. C. 1977. Prognostic efficacy of early clinical categorization of myocardial infarction patients. *Circulation* **56**, 816–819.

Pozen, M. W., D'Agostino, R. B., Mitchell, J. B., Rosenfeld, D. M., Guglielmino, J. M., Schwartz, M. L., Teebagy, N., Valentine, J. M., and Hood, W. B. 1980. The usefulness of a predictive instrument to reduce inappropriate admissions to the coronary care unit. *Ann. Intern. Med.* **92**, 238–242.

Pozen, M. W., D'Agostino, R. B., Selker, H. P., Sytkowski, P. A., Hood, W. B., Jr. 1984. A predictive instrument to improve coronary-care-unit admission practices in acute ischemic heart disease: A prospective multicenter clinical trial. *N. Engl. J. Med.* **310**, 1273–1278.

Reggia, J. A., and Tuhrim, S., eds. 1985. *Computer Assisted Medical Decision Making. Computers in Medicine Series*, Vol. 2. Springer-Verlag, New York.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. 1986. Learning internal representations by error propagation. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, D. E. Rumelhart and J. L. McClelland, eds., pp. 318–364. MIT Press, Cambridge, MA.

Saito, K., and Nakano, R. 1988. Medical diagnostic expert system based on PDP model. In *Proceedings of the International Joint Conference on Neural Networks*, I, 255–262.

Smith, J. W., Everhart, J. E., Dickson, W. C., Knowler, W. C., and Johannes, R. S. 1988. Using the ADAP learning algorithm to forecast the onset of diabetes mellitus. *Symp. Comput. Applic. Med. Care Proc.* **12**, 261–265.

Szolovits, P., Patil, R. S., and Schwartz, W. B. 1988. Artificial intelligence in medical diagnosis. *Ann. Intern. Med.* **108**, 80–87.

Tierney, M. W., Roth, B. J., Psaty, B., McHenry, R., Fitzgerald, J., Stump, D. L., Anderson, F. K., Ryder, K. W., McDonald, C. J., and Smith, D. M. 1985. Predictors of myocardial infarction in emergency room patients. *Crit. Care Med.* **13**, 526–531.

Weigend, A. S., Huberman, B. A., and Rumelhart, D. E. 1990. Predicting the future: A connectionist approach. PDP Research Group Technical Report. *Int. J. Neural Syst.*, submitted.

Widrow, G., and Hoff, M. E. 1960. Adaptive Switching Circuits Institute of Radio Engineering Western Electronic Show and Convention. Convention Record, Part 4, pp. 96–104.

Received 29 August 90; accepted 26 September 90.

2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Evolving a Fuzzy Goal-Driven Strategy for the Game of Geister: An Exercise in Teaching Computational Intelligence

Andrew R. Buck, *Student Member IEEE*, Tanvi Banerjee, *Student Member IEEE*, and James M. Keller, *Fellow IEEE*

***Abstract*—This paper presents an approach to designing a strategy for the game of Geister using the three main research areas of computational intelligence. We use a goal-based fuzzy inference system to evaluate the utility of possible actions and a neural network to estimate unobservable features (the true natures of the opponent ghosts). Finally, we develop a coevolutionary algorithm to learn the parameters of the strategy. The resulting autonomous gameplay agent was entered in a global competition sponsored by the IEEE Computational Intelligence Society and finished second among eight participating teams.**

I. Introduction

GEISTER (German for ghosts) is a strategic board game played between two players on a 6x6 grid. Each player is given four good ghosts and four evil ghosts that are initially placed in any configuration within the 2x4 home areas of each player as shown in Fig. 1. The nature of each ghost (good or evil) is marked on the back of the game piece and is hidden to the opponent player. The players alternate moving their ghosts forward, backward, left, or right (never diagonally) in order to achieve one of three possible victory conditions:

1) Capture all of the opponent's good ghosts
2) Have the opponent capture all evil ghosts
3) Move a good ghost off the board from one of the opponent's corner spaces

An opponent's ghost is captured by moving a ghost onto the same space, at which point its true nature is revealed.

This is a game involving incomplete information. Players do not know with complete certainty which opponent ghosts are good and which are evil. Games of this type differ from games of perfect information such as chess and Go, and require additional strategies to manage the uncertainty. One popular approach is *determinization* [1], in which the uncertain features are randomly sampled from the set of possible values. However, this approach is inefficient and requires a large computational budget to simulate all of the various possibilities. An alternate approach that we use in this paper is to model the uncertain features as fuzzy sets and use a fuzzy inference system to develop the gameplay strategy. This has the advantage of computational speed as well as allowing for a custom, hand-picked set of rules. This is particularly important for developing games on limited hardware such as mobile phones [2].

Our method for developing an autonomous agent to play Geister consists of three parts, utilizing the three main research areas of computational intelligence (CI). It was designed originally as a series of projects for an introductory course on computational intelligence, and as an entry in the "Ghost Challenge 2013" competition organized by the IEEE Computational Intelligence Society. This challenge was issued to promote student involvement and to encourage novel approaches for developing artificial agents using CI techniques.

First, we design a fuzzy inference system that evaluates the utility of possible moves. For inputs, the system uses both observable game state features and estimates of the unknown opponent ghost natures. The outputs are computed using a set of fuzzy rules and are interpreted as the value of a particular ghost pursuing a specific goal. The ghost with the highest valued goal is chosen to be moved in pursuit of that goal.

Second, we use a neural network to perform the estimation of the opponent ghost natures. Observable features such as initial position and movement are used to construct a feature vector for each opponent ghost. A neural network is then trained using a dataset gathered over many games to classify the ghosts as good or evil. During gameplay, the trained neural network is used to compute a good and evil confidence value for each opponent ghost.

Finally, we use a coevolutionary algorithm to improve our hand-built strategy. The membership functions and rules of the fuzzy inference system, along with the weights of the neural network, are encoded in a chromosome structure that represents a particular strategy. A population of these strategies competes and evolves over time, learning a better set of strategy parameters than our initial implementation.

Evolutionary methods such as coevolution have been used by many researchers to learn interesting and unique strategies for games of incomplete or imperfect information [3]. Methods such as *complexification* [4] show that by gradually increasing the complexity of a genetic representation, more elaborate and sophisticated strategies can be developed. We use these ideas to design our algorithm and construct our game-playing agent.

The remainder of this paper is organized as follows. In Section II we discuss the development of the goal-driven fuzzy inference system. Section III covers the design and training of the neural network to estimate the opponent ghost natures. Section IV outlines our coevolutionary algorithm to learn the strategy parameters. The performance of our agent in local play and the global IEEE CIS competition is

A. R. Buck, T. Banerjee, and J. M. Keller are with the Department of Electrical and Computer Engineering, University of Missouri-Columbia, MO 65211, USA (email: arb9p4@mail.missouri.edu; tsbycd@mail.missouri.edu; kellerj@missouri.edu).

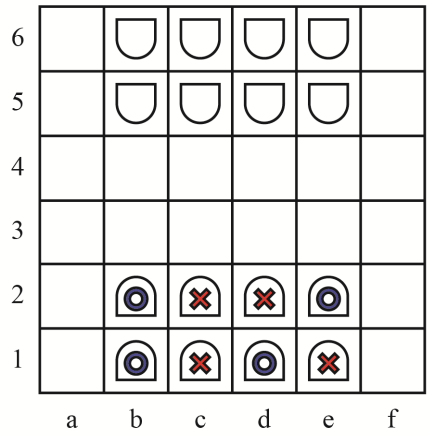


Fig. 1. Initial board position for Geister. Each player has four good ghosts (marked with a blue circle) and four evil ghosts (marked with a red cross). The markings are hidden to the other player so that the true nature of the opponent ghosts is unknown.

discussed in Section V. Lastly, our conclusions and ideas for future work are given in Section VI.

II. Creation of a Fuzzy Inference System

The core decision component of our autonomous agent is a fuzzy inference system (FIS) that can evaluate the utility of possible actions. Each ghost can move in up to four directions resulting in up to 32 possible actions on a player's turn (although, due to board boundaries and the inability to move onto friendly ghosts, the number of actions is typically much smaller). While it may be possible to design a FIS that provides the values of these actions directly, we opted instead to use a goal-driven approach in which each ghost is given a set of goals to pursue. The value of each goal is computed by the FIS for each of the player's ghosts. This allows us to use higher-level features as inputs for the FIS and to more appropriately assess if the output matches the desired behavior.

A. Input Features

Most of the input features for our system are integer measurements, either a distance (in moves) or a number of ghosts. In order to define fuzzy rules based on linguistic expressions, we define linguistic terms for each input feature and a corresponding membership function to map the crisp measurements into fuzzy values. We define a triangular membership function as Tri(a, b, c), where the interval [a, c] is the support and b is the peak of the membership function. Likewise, we define a trapezoidal membership function as Trap(a, b, c, d), where the interval [a, d] is the support and the interval [b, c] is the core of the membership function. The full list of input features is given in Table I.

The first set of input features describe the current state of the game and are the best indication of who is currently winning.

1) Captured Good (CG): The number of good opponent ghosts that have been captured.
2) Captured Evil (CE): The number of evil opponent ghosts that have been captured.
3) Lost Good (LG): The number of good ghosts that have been captured by the opponent.
4) Lost Evil (LE): The number of evil ghosts that have been captured by the opponent.

Each of these features is an integer in the range [0, 3]. If all four good ghosts or all four evil ghosts are lost by either player, the game is over.

The second set of input features applies to a specific ghost

TABLE I
Input Features to the Fuzzy Inference System

Input Linguistic Variable	*Term*	*Membership Function*
Captured Good (CG)	Low (L)	Tri(0, 0, 3)
	High (H)	Tri(0, 3, 3)
Captured Evil (CE)	Low (L)	Tri(0, 0, 3)
	High (H)	Tri(0, 3, 3)
Lost Good (LG)	Low (L)	Tri(0, 0, 3)
	High (H)	Tri(0, 3, 3)
Lost Evil (LE)	Low (L)	Tri(0, 0, 3)
	High (H)	Tri(0, 3, 3)
Closest Ghost Distance (CGD)	Adjacent (A)	Tri(0, 1, 2)
	Near (N)	Trap(0, 0, 2, 6)
	Far (F)	Trap(2, 6, 10, 10)
Closest Ghost Good Confidence (CGGC)	Low (L)	Tri(0, 0, 1)
	High (H)	Tri(0, 1, 1)
Closest Ghost Evil Confidence (CGEC)	Low (L)	Tri(0, 0, 1)
	High (H)	Tri(0, 1, 1)
Distance to Opponent Exit (DOE)	Adjacent (A)	Tri(0, 1, 2)
	Near (N)	Trap(0, 0, 2, 6)
	Far (F)	Trap(2, 6, 8, 8)
Exit Congestion (EC)	Low (L)	Tri(0, 0, 8)
	High (H)	Tri(0, 8, 8)
Opponent Distance to Home Exit (ODHE)	Adjacent (A)	Tri(0, 1, 2)
	Near (N)	Trap(0, 0, 2, 7)
	Far (F)	Trap(2, 7, 8, 8)
Distance to Home Exit (DHE)	Adjacent (A)	Tri(0, 1, 2)
	Near (N)	Trap(0, 0, 2, 4)
	Far (F)	Trap(2, 4, 10, 10)
Others Distance to Home Exit (OTDHE)	Adjacent (A)	Tri(0, 1, 2)
	Near (N)	Trap(0, 0, 2, 7)
	Far (F)	Trap(2, 7, 8, 8)

TABLE II
Output Goals of the Fuzzy Inference System

Output Goal	*Term*	*Membership Function*
Capture Closest Ghost (CAP)	Low (L)	Tri(0, 0, 1)
	Medium (M)	Tri(0.1, 0.5, 0.9)
	High (H)	Tri(.6, 1, 1)
Block (BLOCK)	Low (L)	Tri(0, 0, 1)
	Medium (M)	Tri(0.1, 0.5, 0.9)
	High (H)	Tri(.6, 1, 1)
Exit (EXIT)	Low (L)	Tri(0, 0, 1)
	Medium (M)	Tri(0.1, 0.5, 0.9)
	High (H)	Tri(.6, 1, 1)
Escape (ESC)	Low (L)	Tri(0, 0, 1)
	Medium (M)	Tri(0.1, 0.5, 0.9)
	High (H)	Tri(.6, 1, 1)
Tempt Opponent Ghost (TEMPT)	Low (L)	Tri(0, 0, 1)
	Medium (M)	Tri(0.1, 0.5, 0.9)
	High (H)	Tri(.6, 1, 1)

and measures the distance to the closest opponent ghost and its estimated nature. We consider only the closest opponent ghost for simplicity.

5) Closest Ghost Distance (CGD): The distance to the closest opponent ghost.
6) Closest Ghost Good Confidence (CGGC): The confidence that the closest ghost is good.
7) Closest Ghost Evil Confidence (CGEC): The confidence that the closest ghost is evil.

The next set of input features also applies to a specific ghost and is used to determine how easily the ghost could win the game by exiting from one of the opponent corners.

8) Distance to Opponent Exit (DOE): The distance to the closest opponent corner exit.
9) Exit Congestion (EC): The number of opponent ghosts in the rectangular region between this ghost and the closest opponent corner exit.

The final set of input features is used to determine how close the opponent is to escaping from a home corner exit and how difficult it would be to block.

10) Opponent Distance to Home Exit (ODHE): The fewest number of moves an opponent ghost would need to exit from a home corner.
11) Distance to Home Exit (DHE): The distance between this ghost and the home corner nearest to an opponent ghost.
12) Others Distance to Home Exit (OTDHE): The number of our own ghosts that are closer than this ghost to blocking the home corner exit nearest to an opponent ghost.

B. Output Goals

The outputs of the FIS represent the utility values of different goals that a ghost can pursue. These are represented in the normalized range [0, 1] and are assigned linguistic terms for use in the fuzzy rules. The membership functions of the following goals are given in Table II.

1) Capture Closest Ghost (CAP): Move this ghost toward the nearest opponent ghost and capture it.
2) Block (BLOCK): Move this ghost toward one of the home corner exits to block an opponent ghost from escaping.
3) Exit (EXIT): Move this ghost toward one of the opponent's corner exits and off the board.
4) Escape (ESC): Move this ghost away from any opponent ghosts to avoid being captured.
5) Tempt Opponent Ghost (TEMPT): Move this ghost toward a space adjacent to an opponent ghost with the hope that it will become captured.

C. Rules

We design a set of rules to indicate under which circumstances each of the output goals is a useful pursuit. The "Capture Closest Ghost" and "Block" goals can be applied to both good and evil ghosts, however the "Exit" and "Escape" goals apply only to good ghosts and the "Tempt" goal applies only to evil ghosts. These restrictions are not strictly necessary, as it may be a useful bluffing strategy for a good ghost to imitate an evil ghost or vice versa. We allow for this flexibility in the final evolutionary training of our method.

The rules for the goals are given in Tables III-VII. Each row in a rule table represents a separate rule. These were hand-picked to represent a reasonable initial strategy for the game of Geister. Along with the membership functions defined for the input and output features, this comprises a Mamdani Fuzzy Inference System. To determine the output value of a particular goal, the crisp input feature values are fuzzified according to the defined membership functions. The minimum firing strength of each antecedent in the rule determines the maximum value of the consequent membership function. The consequent membership functions are summed together and defuzzified into a crisp value using the "mean of maximum" defuzzification method.

TABLE III
RULES FOR THE GOAL "CAPTURE CLOSEST GHOST"

Inputs				***Outputs***
CE	CGD	CGGC	CGEC	CAP
H				L
			H	L
L	N	¬L		M
L	A	H		H

TABLE IV
RULES FOR THE GOAL "BLOCK"

Inputs			***Outputs***
ODHE	DHE	OTDHE	BLOCK
F			L
N	N	L	H
N	N	M	L
N	N	H	L

TABLE V
RULES FOR THE GOAL "EXIT"

Inputs					***Outputs***
CG	CE	LG	DOE	EC	EXIT
				H	L
L	H				M
		L	N	L	M
			A		H

TABLE VI
RULES FOR THE GOAL "ESCAPE"

Inputs		***Outputs***
LG	CGD	ESC
L	F	L
	N	M
H	A	H

TABLE VII
RULES FOR THE GOAL "TEMPT"

Inputs		***Outputs***
LE	CGD	TEMPT
L	F	M
H		M
H	A	H

D. Determining an Action

After applying the FIS to all ghosts to get the value of each goal, we must now determine which ghost to move. The goals for a ghost each have a location on the board. For the goal "Capture Closest Ghost," the location is the space of the closest opponent ghost. For "Block," the location is the home corner space closest to an opponent ghost. For "Exit," the location is the closest opponent corner exit. For "Escape" and "Tempt," there may be multiple goal locations. Any adjacent space with no threat of being captured on the next turn serves as a location for the "Escape" goal. The four spaces adjacent to the closest opponent ghost work as locations for the "Tempt" goal, provided that they are not already occupied by one of our own ghosts.

The goal locations for each ghost are assigned to zero, one, or two possible movement directions. If the goal is directly above, below, to the left, or to the right of the ghost, the value of that goal is added to the value of movement in that direction. If the goal is at an angle, the two directions on either side of the goal location each receive half of the goal's value. If any of the movement directions are not valid moves, such as moving off the board (except in the case of exiting from an opponent's corner) or moving onto one of our own ghosts, the value of movement in that direction is fixed to zero.

After accumulating the values of the goals for each ghost in terms of movement direction values, the list of possible moves is sorted from most valuable to least valuable. We use a probabilistic selection strategy, selecting the best possible move 80% of the time, and moving on to the next possible move the rest of the time. This move again has an 80% chance of being chosen and the process repeats until a move is picked or the list is exhausted. This approach adds a random element to our strategy making it more difficult for the opponent to guess our moves. It also helps to prevent infinite loops when playing against itself or a similar strategy.

III. Profiling the Nature of an Opponent Ghost Using a Neural Network

All of the input features in the previous section can be measured through direct observation except two: "Closest Ghost Good Confidence" and "Closest Ghost Evil Confidence." Because the true natures of the opponent ghosts are hidden, these values must be estimated. Our approach is based on the work done by Aiolli and Palazzi in [5]. In their work, they record a set of 17 features for each opponent ghost over the course of an entire game. These are then used to build a nearest neighbor classifier giving the predicted nature of the opponent ghost.

We use the same set of features as described in [5], but with a Neural Network to estimate the confidence that a ghost is good or evil. The first eight features are binary values corresponding to the initial position of the opponent ghost. Each ghost has a different one of these features that is non-zero. The next two features are also binary and indicate if the ghost was the first or second piece to be moved. The next three features are integer valued and indicate how many forward, backward, and lateral moves the ghost made. The final four features count the different ways the ghost responds to being threatened, either by capturing an opponent ghost, escaping to a safe location, moving to threaten another ghost, or not moving at all.

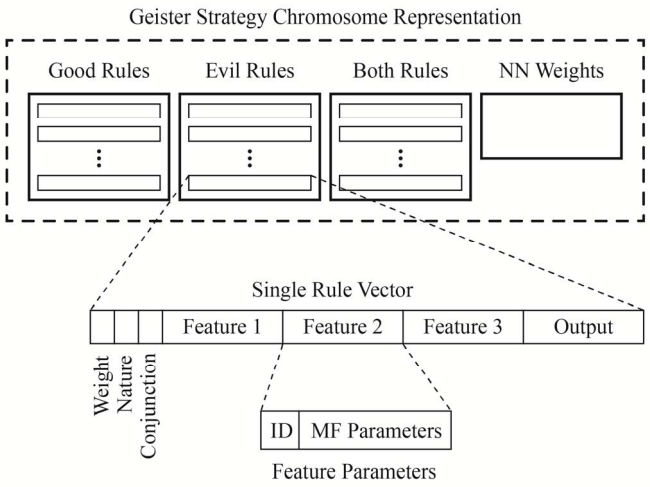


Fig. 2. Chromosome representation of a strategy. Each chromosome contains a rule base for good ghosts, evil ghosts, and one for both, along with a set of neural network weights. The rule bases contain rule vectors that describe up to three input features and a single output goal. The features are represented with an integer ID and four parameters for a trapezoidal membership function.

These features are updated and collected throughout the course of the game. The final feature values for each ghost are saved at the end of each game resulting in eight new sample vectors: four good ghost samples and four evil ghost samples. After playing 100 games with the hand-built strategy described above playing against itself, we obtain a training dataset of 800 samples with half representing good ghosts and half representing evil ghosts. During these training games, the good and evil confidence values are fixed at 0.5, ignoring the nature estimation step. While it would have been helpful to build the training dataset from games played between human players or more advanced AI players, this was not feasible at the time. We instead compensate for this deficiency by updating the weights of the neural network during the evolutionary learning phase described in the next section.

We construct a multi-layer perceptron with 10 nodes in the hidden layer, 17 inputs, and 2 outputs corresponding to the good and evil confidence values. This network is then trained on the collected dataset using the backpropagation algorithm. Once the network weights are obtained, new features from an opponent ghost sampled during gameplay can be passed through the network to obtain the confidence values that the ghost is good or evil.

IV. Learning Parameter Values with Coevolution

The strategy we have developed thus far has been based on the hand-picked rules and membership functions that make up the fuzzy inference system. While this represents a good starting point, there is still room for improvement using computational optimization techniques. We design an algorithm based on coevolution that allows multiple versions of our strategy to compete with one another in order to move

toward the optimal strategy. After many iterations of the algorithm, the strategies evolve to become more competitive than our original implementation.

A. Representation

First, we need to encode our FIS and neural network as a parameterized chromosome. An overview of our representation is given in Fig. 2. We represent the FIS as three separate rule bases: one containing rules for good ghosts, one with rules for evil ghosts, and one with rules that are applied to both. Each rule base contains a variable number of rules, with each rule taking the form of a 23-parameter vector.

The first parameter in a rule vector is a floating-point number between 0 and 1 indicating the weight of the rule. The consequent of the rule is multiplied by this value to modify its importance. The second parameter is an integer indicating if the rule is in the good, evil, or both rule bases. This is used during the initialization and mutation procedures. The third parameter is a Boolean value indicating if the conjunction for the rule should be "And" or "Or."

The remaining 20 parameters of a rule vector are grouped into four sets of five values representing the membership functions of up to three input features and one output goal. Each of these sets contains one integer parameter indicating which feature or goal the membership function applies to and four floating-point values used as parameters for a trapezoidal membership function. The domains of each feature are normalized into the range [0, 1], and the membership functions (defined for our hand-built strategy in Tables I and II) are explicitly defined in each rule. The floating-point values must be sorted in ascending order, and can also represent triangular membership functions if the two middle values are equivalent. The integer parameter for an input feature may be zero to indicate an absence of that feature, in which case the four membership function parameters are ignored. This produces a rule with fewer than three antecedents; however, each rule must have at least one antecedent and an output goal. There must also be at least one rule for good ghosts and at least one rule for evil ghosts among the three rule bases.

The parameters of the neural network are stored in a straightforward manner. The network has 17 inputs, 10 hidden nodes, and 2 outputs. Including the bias terms, this gives 180 values between the input and hidden layers and 22 values between the hidden and output layers for a total of 202 floating-point parameters. These are stored in tables, but they can be linearized into a single 202-element vector for the purposes of initialization, crossover, and mutation.

Algorithm I: Coevolutionary Algorithm

```
/* Initialization */
Initialize population P with popSize copies of our
   hand-built strategy
For i = 2 to popSize do
   P_i = mutate(P_i)
End For

generation = 0
While stopping criteria not met

   /* Evaluate Fitness */
   For i = 1 to popSize do
      fitnessValues_i = 0
      For j = 1 to numOpponents do
         Opponent = random strategy from P
         wins = 0; losses = 0; draws = 0
         Play numGames games against Opponent
         Update wins, losses, and draws
         score = 2*wins – 2*losses – draws
         fitnessValues_i += score
      End For
   End For

   /* Create Next Generation */
   P^new = {}
   Sort P by fitnessValues in descending order
   Copy first eliteSize strategies from P into P^new
   While |P^new| < |P|
      Select two parents p_1 and p_2 from P using
         tournament selection with a tournament size of t
      With probability crossoverProb, create children:
         [c_1, c_2] = crossover(p_1, p_2)
      With probability mutationProb, mutate children:
         c_1 = mutate(c_1); c_2 = mutate(c_2)
      Add c_1 and c_2 into P^new
   End While
   generation++

   /* Maintain Diversity */
   If generation is a multiple of restartRate then
      Replace last restartSize elements of P^new with new
         random strategies
   End If

End While
Return population of evolved strategies P
```

B. Initialization

The initial population for our evolutionary algorithm consists of several mutated copies of our hand-built strategy, along with an unmodified version. As the algorithm progresses, new randomly-built strategies are needed to maintain diversity and promote exploration. We create a random strategy by adding a set number of random rules (about 20) to the rule bases and by generating a set of random weights for the neural network.

Each parameter in a rule has a bounded domain, and a random rule is created by selecting each parameter from a uniform distribution over the appropriate domain. The second parameter in each rule indicates which rule base the rule belongs to. We must sort the four membership function parameters for each input feature and the output goal, and we must also ensure that each rule has at least one input antecedent. Additionally, there must be at least one rule for good ghosts and at least one rule for evil ghosts across the three rule bases.

C. Crossover

The purpose of crossover in an evolutionary algorithm is to recombine the information in two or more parents to produce offspring that share information from all of the parents. We perform crossover between two chromosome strategies by exchanging their rules and neural network parameters. Complete rules are exchanged without modification, whereas the weights in the neural network are considered individually.

Two child strategies are created initially as copies of the parents. We iterate over all of the rules from each rule base and with a certain crossover probability (80% in our method), each rule is given to the other child strategy. Likewise, we iterate over each parameter in the neural networks of the two strategies and exchange the weights using the same crossover probability.

D. Mutation

Mutation is applied to a single chromosome strategy in order to vary the parameters and explore new regions of the search space. With a certain mutation probability (10% in our method), each parameter in the rule bases and neural networks is changed to a new random value. For rule parameters that are integers, a new value is chosen within the appropriate domain. For real-valued parameters in the rules and neural networks, a small amount of Gaussian noise (σ = 0.2) is added to the value. For rule parameters with a bounded domain, this value is clipped to remain in-bounds. After selecting new values, the membership function parameters must be resorted.

E. Coevolutionary Learning

Our coevolutionary algorithm is outlined in Algorithm I and the specific parameters we used are given in Table VIII. We begin by creating a population of different strategies, all based on mutations of our original hand-built strategy. We also include the original unmodified strategy in the initial population. As the algorithm progresses, we compare the original strategy to the best strategy discovered in order to gauge how much learning has occurred.

Each generation of the algorithm starts by evaluating the fitness of each individual. A coevolutionary algorithm computes a relative fitness for each individual based on how it compares to other individuals. One strategy commonly used is to select a random sample from the population and evaluate the relative performance [6]. We select 10 opponents randomly with replacement from the population and play 20 games with each one, alternating as the first and second player. This results in 200 games of Geister being played for each fitness evaluation. We count the number of wins, losses, and draws for a strategy (a game is considered a draw if neither player has won after 100 moves), and compute the fitness as

$$fitness = 2*wins - 2*losses - draws. \quad (1)$$

It should be noted that due to the random nature of the fitness evaluation, a strategy's fitness can change between generations. This makes it difficult to keep track of the best solution found during the evolutionary learning process.

TABLE VIII
COEVOLUTIONARY ALGORITHM PARAMETERS

Parameter	*Value*	*Description*
popSize	20	Population size
numOpponents	10	Number of opponents each strategy plays against to determine fitness
numGames	20	The number of games played between opponents
eliteSize	2	Number of elite solutions that automatically survive each generation
t	2	Tournament size for selection
crossoverProb	0.8	Crossover probability
mutationProb	0.1	Mutation probability
σ	0.2	Spread of Gaussian noise for mutation
restartRate	10	Frequency that new random strategies are added to the population
restartSize	10	Number of random strategies to add with each restart

Nevertheless, we use the concept of elitism and automatically copy the two best individuals into the next generation without modification. A common approach in coevolutionary algorithms is to use a "Hall of Fame" in which the best individuals of the population are saved in an elite set, which must compete with every other individual in the population during the fitness evaluation [7]. We use elitism in our algorithm, but not a hall of fame. Including this modification could improve the performance of our algorithm in future experiments.

After copying the elite individuals into the new population, the remainder of the new population is created by selecting parents and creating offspring. We use the computed fitness values with tournament selection and a tournament size of two. This gives a reasonable selective pressure that tends to pick better individuals for reproduction, but allows lower performing strategies to reproduce as well. Then, with a probability of 0.8 we perform crossover, resulting in two children. We perform mutation on the children with a probability of 0.1 and add the children into the new population.

To maintain diversity, we replace the lower half of the population with new random strategies every 10 generations. This helps restart the algorithm in new areas of the search space and keeps the population from converging to a single strategy. The algorithm can be run for as long as time will allow, after which the top strategies in the population are returned.

V. PERFORMANCE RESULTS

A. Ghosts Challenge

Our game-playing agent was entered into the "Ghosts Challenge 2013" competition organized by the IEEE Computational Intelligence Society. The competition was open to all student members of the IEEE CIS and encouraged the use of computational intelligence techniques for developing autonomous agents. Our agent was submitted under the team name "mutigers" and competed against seven other submitted agents.

In developing our agent for the competition, we were able to run the coevolutionary algorithm described in the previous

TABLE IX
FINAL RANKINGS OF THE IEEE CIS GHOST CHALLENGE 2013

Ranking	*Team*	*Points*	*Rounds Difference*	*Matches Won*	*Matches Lost*	*Matches Drawn*	*Rounds Won*	*Rounds Lost*	*Rounds Drawn*	*Game Types*[a]						
										WC	*WL*	*WE*	*LC*	*LL*	*LE*	*D*
1	BLISS	21	209	7	0	0	274	65	11	172	17	85	0	52	13	11
2	mutigers	18	207	6	1	0	274	67	9	8	9	257	39	16	12	9
3	Eliot_CS7750	15	116	5	2	0	232	116	2	0	4	228	38	3	75	2
4	Skynet	12	9	4	3	0	172	163	15	15	0	157	49	18	96	15
5	FightGhost	9	37	3	4	0	190	153	7	10	21	159	19	9	125	7
6	RST	6	-184	2	5	0	35	219	96	7	21	7	16	12	191	96
7	WAIYNE1	1	-196	0	6	1	21	217	112	2	19	0	20	3	194	112
8	tsengine	1	-198	0	6	1	25	223	102	0	25	0	33	3	187	102

[a]Game types lists the distribution of the seven different ways a game can end for each team. The types are: WC (win by capturing all good ghosts of the opponent); WL (win by losing all evil ghosts); WE (win by exiting from an opponent corner with a good ghost); LC (lose by having all good ghosts get eaten); LL (lose by capturing all evil ghosts of the opponent); LE (lose by allowing the opponent to exit from a home corner with a good ghost); and D (draw).

section for two days. This was done using Matlab code running in parallel on a six-core i7 machine at 3.4 GHz with Windows 7. In this time, our algorithm evolved 87 generations of strategies. The final submission was rewritten in Java, as this was required to interface with the competition server. We included the top three evolved strategies and our original hand-drafted strategy in our competition agent. Our agent begins a match by playing with the best evolved strategy. After losing twice in a row, a different strategy is selected at random from the other strategies.

The competition was organized as a round-robin tournament in which each team played a match of 50 games against every other team. In addition to the 100 move limit imposed per game, agents have only 10 seconds to make each move before the server enforces a random choice. The teams were awarded 3 points for winning the most games in a match, 1 point for tying, and 0 points for losing. In the case of a tie in points, the winner is decided by the difference between the number of rounds won and lost. The final rankings of the competition are given in Table IX.

Our team, mutigers, placed second in the competition, losing only to the top scoring team, BLISS. Upon inspection of the game logs, it was discovered that mutigers, along with most of the other successful teams, preferred to attempt victory by moving a good ghost off the board from an opponent exit. However, the winning team, BLISS, favored capturing the opponent's good ghosts. Our agent was not able to successfully defend our good ghosts from BLISS to prevent them from being captured.

The implementation details of the competing agents were not released at the time of this writing, so it is difficult to make definitive conclusions about the different strategies. However, the distributions of the game types for each agent give some indication of the strategies involved. Clearly our strategy prefers to move a good ghost to the exit over capturing other ghosts. This could be due to low confidence values in our nature estimation, or a bias in our fuzzy rule bases toward the "Exit" goal. This shows the importance of having a diverse set of strategies that can counter different opponent tactics.

B. Evolving a Strategy

Due to the time constraints of the competition, the strategies we employed came from just 87 generations of evolution. We allowed the algorithm to continue running to 350 generations over the course of 10 days to observe the effect of additional learning time. The maximum and mean fitness scores for each generation are plotted in Fig. 3. Overall, these plots are very noisy showing wide fluctuations in the fitness values between generations. This occurs partly because each fitness evaluation involves playing a set of games against a random selection of opponents, and is therefore not a stable measure. Additionally, as the algorithm progresses the population as a whole becomes more competitive. This makes it more difficult for the best strategies to continue winning and allows new strategies to take the top spot. The plots show a hint of a periodic trend as a good strategy rises with a high fitness score, and then creates variations of itself through reproduction, creating more challenging competition. The initial rise during the first 30 or so generations shows the greatest amount of improvement as the population moves away from strategies based solely on our hand-drafted approach. The mean fitness value of the population stays roughly the same as the improvements learned through coevolution are balanced by the randomly introduced strategies every 10 generations, maintaining a diverse population.

To verify that the strategies in the population are improving, we play a set of games with the top strategy from a given generation against the entire population of another generation. The results of playing a match of 50 games with each of the 20 strategies in a population every 50 generations is given in Fig. 4. The colors in the matrix indicate the fitness scores obtained using Equation 1. Each row in the matrix represents the performance of the best strategy from a generation. The columns represent which generation's population the strategy had to play against.

As expected, the lower rows of the matrix are lighter, indicating higher fitness scores for the best strategies of later generations. Our expectation is that the lower-left corner should be light, indicating that the best strategy of the latest generation performed very well against the original population. Likewise, the upper-right corner should be dark to indicate that the original strategy did poorly against the most evolved population. In general this is true, but these are not the lightest or darkest cells in the matrix. We see a peak in the maximum fitness plot of Fig. 3 around 100 generations and this correlates to a light row in the matrix of Fig. 4. This was a rather good strategy, as it performed reasonably well against the other populations. We also see a dark column in

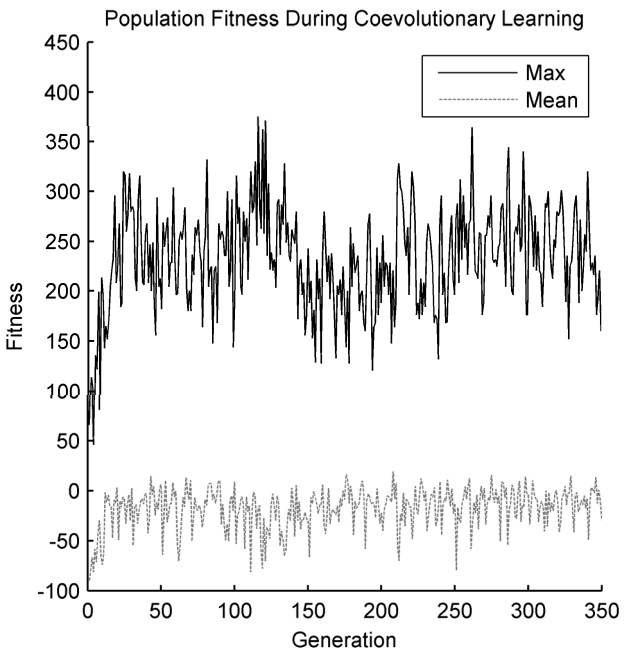


Fig. 3. Population fitness during coevolutionary learning. The fitness of each individual in the population of 20 strategies is obtained by playing 20 games against 10 random individuals and computing a score based on Equation 1.

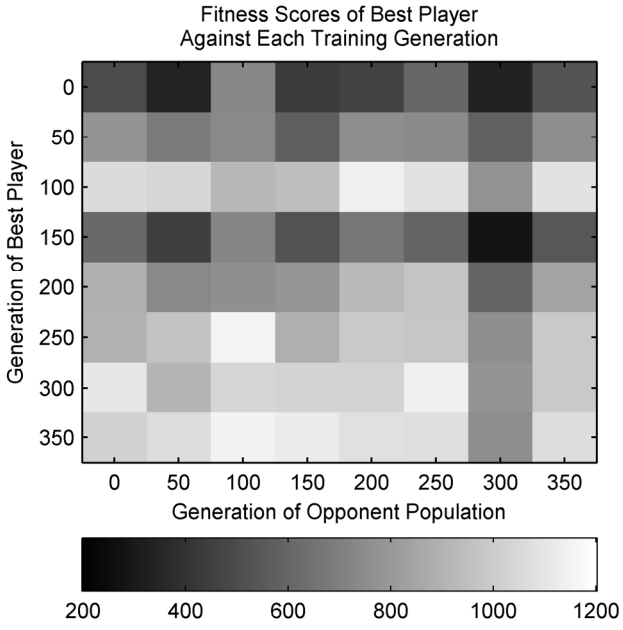


Fig. 4. Fitness scores of the best strategy every 50 generations playing against the entire population from every 50 generations of training. The best strategies from each generation played 50 games against all 20 individuals from the other generations. The fitness scores were computed using Equation 1.

the 300th generation indicating a weak population. These fluctuations in fitness seem to be common for coevolutionary algorithms with relative fitness measures.

VI. Conclusion and Future Work

Our intent with this project was to design a competitive agent for Geister using CI techniques that could be used as class projects. Although our method did not win the Ghost Challenge 2013 competition, our experiments demonstrate the applicability of coevolutionary learning using a fuzzy inference system and a neural network. Our goal-driven approach allows high-level strategies to be defined in terms of fuzzy rules and improved upon with evolutionary methods.

One improvement that would help our algorithm greatly is more training data. The neural network weights are learned initially from only our hand-drafted strategy. Multiple strategies from other sources, such as other teams in the competition or human players, would improve the quality of the training set. Additionally, our algorithm could be modified to update the weights of the neural network over the course of several games. This would allow it to adapt to various strategies on the fly and increase the accuracy of the ghost nature estimations.

Finally, our strategy could be improved by utilizing a look-ahead feature to determine which actions will be more beneficial in the future. This is done automatically by traditional game playing AI such as mini-max search or Monte Carlo Tree Search. A comparison with these methods could reveal additional strengths and weaknesses and result in a more globally competitive game playing agent.

References

[1] D. Whitehouse, E. J. Powley, and P. I. Cowling, "Determinization and information set Monte Carlo Tree Search for the card game Dou Di Zhu," in *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*, 2011, pp. 87–94.

[2] F. Aiolli and C. E. Palazzi, "Enhancing artificial intelligence on a real mobile game," *Int. J. Comput. Games Technol.*, vol. 2009.

[3] S. Lucas and G. Kendall, "Evolutionary computation and games," *Comput. Intell. Mag.*, vol. 1, no. 1, pp. 10–18, Feb. 2006.

[4] K. O. Stanley and R. Miikkulainen, "Competitive coevolution through evolutionary complexification," *J. Artif. Intell. Res.(JAIR)*, vol. 21, pp. 63–100, 2004.

[5] F. Aiolli and C. Palazzi, "Enhancing artificial intelligence in games by learning the opponent's playing style," in *Proceedings of the IFIP-ECS Conference*, 2008, pp. 1–10.

[6] J. Reed, R. Toombs, and N. A. Barricelli, "Simulation of biological evolution and machine learning," *J. Theor. Biol.*, vol. 17, no. 3, pp. 319–342, 1967.

[7] C. D. Rosin and R. K. Belew, "New methods for competitive coevolution.," *Evol. Comput.*, vol. 5, no. 1, pp. 1–29, Jan. 1997.

The Collateral Damage of Internet Censorship by DNS Injection *

Sparks
Hovership Nebuchadnezzar
Zion Virtual Labs
zion.vlab@gmail.com

Neo†
Hovership Nebuchadnezzar
Zion Virtual Labs
zion.vlab@gmail.com

Tank
Hovership Nebuchadnezzar
Zion Virtual Labs
zion.vlab@gmail.com

Smith
Hovership Nebuchadnezzar
Zion Virtual Labs
zion.vlab@gmail.com

Dozer
Hovership Nebuchadnezzar
Zion Virtual Labs
zion.vlab@gmail.com

ABSTRACT

Some ISPs and governments (most notably the Great Firewall of China) use DNS injection to block access to "unwanted" websites. The censorship tools inspect DNS queries near the ISP's boundary routers for sensitive domain keywords and injecting forged DNS responses, blocking the users from accessing censored sites, such as `twitter.com` and `facebook.com`. Unfortunately this causes large scale collateral damage, affecting communication beyond the censored networks when outside DNS traffic traverses censored links. In this paper, we analyze the causes of the collateral damages comprehensively and measure the Internet to identify the injecting activities and their effect. We find 39 ASes in China injecting forged replies even for transit DNS traffic, and 26% of 43,000 measured open resolvers outside China, distributed in 109 countries, may suffer some collateral damage. Different from previous work, we find that most collateral damage arises from resolvers querying TLD name servers who's transit passes through China rather than effects due to root servers (F, I, J) located in China.

Categories and Subject Descriptors

C.2.0 [**Computer Communication Networks**]: General

General Terms

Measurement, Security

Keywords

DNS, packet injection, Internet measurement, Internet censorship, Great Firewall of China, collateral damage

1. INTRODUCTION

Since DNS is essential for effectively all communication, it is a common target for censorship systems. The most popular approach involves packet injection: a censorship system observes DNS requests and injects fake replies to block communication. Yet censorship systems may affect more than just the censored network.

As a concrete example, consider a query for `www.epochtimes.de` from a US user, using a US-based DNS resolver. The US resolver will need to contact one of the DNS TLD authorities for `.de`, located in Germany. If the path to the selected TLD authority passes through China, then the Chinese Great Firewall will see this query and inject a reply which the US resolver will accept, cache, and return to the user, preventing the user from contacting the proper web server.

Packet injection's popularity as a censorship mechanism arises from its ease of implementation. The censor needs to only monitor traffic and inject responses. Thus network operators have used TCP packet injection to block Peer to Peer traffic [4] or undesirable web content [3], and the Chinese Great Firewall and others use DNS packet injection to block entire sites. While some ISPs are content to block users inside their network from accessing "unwanted" websites using DNS injection, they may not know that their DNS injecting activities potentially affect users outside their network. In the motivating example of contacting `www.epochtimes.de` from the US, the *collateral damage* was due solely to the DNS request passing through a censored network as traceroute verified that the path for HTTP traffic did not pass through a censored network.

Although the DNS community has perceived such collateral damage, they only found it happened when resolvers outside contacted DNS authorities inside the censored country [1], with the most famous examples involving queries from Chile that found themselves routed to the Chinese I-root server [6].

However, the range of the potential damage is actually much more complicated. We find that even querying name servers unrelated to censored countries, resolvers outside could still suffer from collateral damage caused by DNS injection activities from censored transit networks.

In this paper, we make a comprehensive study of the collateral damage caused by DNS injection. Specifically, we try to answer the following three questions:

- How does this collateral damage occur?
- Which ISPs are adopting DNS injection?
- What names and resolvers are affected?

*We use pseudonyms to protect the authors.

†Corresponding author.

For the first question, we analyze the cause from the diversity of DNS resolution paths, as well as the dynamic routing. We utilize two tools, *HoneyQueries* to detect affected paths and *TraceQueries* to detect the point of injection. This enables us to identify the censored ASes. Finally, we perform measurements using *StepNXQueries* which allow us to detect whether a resolver's path to the authorities for the root or a given TLD experience censorship. A survey of 43,842 non-censored resolvers showed 11,579 suffering from some collateral damage. Unlike the results in [1], we find that the most common source of pollution exists on the path between the resolvers and the TLD authorities, particularly the paths to `.de` and `.kr` authorities.

The rest of the paper is organized as follows. In § 2 we give a brief introduction to DNS resolution and how packet injection can disrupt the process. Then we analyze the cause for the collateral damage caused by DNS injection in § 3. In § 4 we describe our experiment methodologies and present the experiment results. We have a discussion in § 5 before concluding in § 6.

2. BACKGROUND

The standard DNS resolution process [8, 9, 5] consists of several pieces, including the stub resolver on the user's computer, the recursive resolver, the root servers ("."), Top Level Domain (TLD) authorities, and the site's authority nameservers as illustrated in Figure 1. When a user generates a request to the recursive resolver, and the resolver has no valid cache information, it first directs that question in full to a root server, which redirects the resolver to the TLD authorities, which redirect to the final authority servers. In the process the resolver caches the intermediate information as well as the final answer.

If an attacker, be it a hacker, an ISP, or a government, can monitor any of the links and inject packets, he can launch a DNS injection attack, replying with a forged response which has the appropriate query question and protocol identifiers but with a bogus DNS answer, mapping the queried domain to either an invalid IP address or an IP address controlled by himself. In the absence of DNSSEC validation, the resolver will generally accept the faked answer because it arrives earlier than the real one, and, as a result, the access to the sensitive site will be blocked or redirected.

The ease of this attack makes it naturally an effective censorship mechanism. It is well known that the Great Firewall of China (GFC) uses this mechanism. The survey of [7], in which the authors queried $>$ 800 DNS resolvers in China, found that 99.88% of them were affected by the GFC.

The collateral damage of GFC was first discussed in a DNS operation mailing list when a Chilean operator found that queries from Chile and California to I.RootServer.NET sometimes experienced DNS pollution [6]. In [1], Brown et. al. analyzed this incident and determined that this kind of pollution could affect many countries because three root DNS server nodes (F, I, and J) have anycast instances in China. They believed that after Netnod withdrew the anycast routes for the Chinese I-root nameserver from CNNIC, the collateral damage should disappear.

Yet there exists an additional collateral damage mechanism. Resolvers only rarely query the DNS root as the root's responses are broad and long lived, lasting in the cache. Yet resolvers must frequently query the TLD authorities. Thus the paths from the resolver to the TLD authorities is as critical as the path from the resolver to the roots.

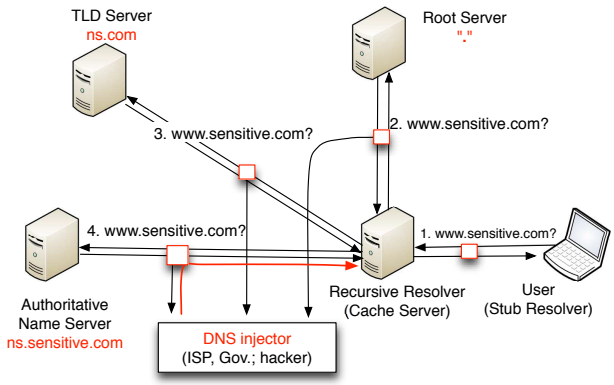


Figure 1: DNS query process and DNS injection

3. CAUSES OF COLLATERAL DAMAGE

Collateral damage occurs when a DNS query from a recursive resolver enters a censored network, causing the censorship mechanism to react. Although intuition would suggest that this would be a rare occurrence, there exist several factors which may cause the censor to receive and react to DNS queries from outsiders.

Iterative Queries: A recursive resolver does not send limited queries, such as asking the root for just the nameservers of the desired TLD. Instead, if it lacks cache entries for the TLD authorities, it sends the *entire* query to a root server. Similarly, the resolver sends the entire query to a TLD authority if there are no cache entries for the domain's authority.

This may be further complicated by "out-of-bailiwick" glue records. Suppose the DNS authorities for `example.com` are `ns1.example.net` and `ns2.example.net`. In the absence of cached data, the resolver will first query for `www.example.com` to a root server and then to a `.com` TLD authority. The reply from the `.com` TLD will now cause the resolver to first query for `ns1.example.net` before resuming the query for `www.example.com`. Thus the resolver will query for `www.example.com` three times: to a root, to a `.com` TLD server, and to `ns1.example.net`, and at least two queries for `ns1.example.net`: to a root and to a `.net` TLD server[1]. Thus a simple "lookup" may generate numerous queries, the disruption of any by censorship would cause resolution to fail.

Redundant Servers and Anycast: Most DNS deployments use multiple servers in multiple networks to increase reliability [2], and actual selection of particular authorities by a recursive resolver is a complex topic, with nameservers using various algorithms. Thus, with 13 different roots and 13 global TLD servers for `.com`, a resolver may experience collateral damage if a path to any one of these 26 IPs passes into a censored network.

Further complicating the picture is the use of *anycast* [10] DNS authorities, where a single IP address may represent a widely deployed system of servers. Two resolvers in different networks may reach different physical servers, along very

[1]If the authority for `example.net` is `ns1.example.net`. Otherwise it can generate even more requests

different paths, even though they are attempting to contact the same IP address.

Censored Transit and Dynamic Routing: The paths from the resolver to the authorities is dynamic, routing through a series of Autonomous Systems (AS), independent networks which together form the Internet. If one transit AS implements censorship, then all traffic which passes through that AS experiences censorship, even if both the source and destination are in non-censored networks. Routing changes also make it difficult to predict when and where DNS queries will pass through censored transit networks.

4. MEASUREMENT AND RESULTS

By measuring of the effect of DNS injection, we want to answer the following two questions related to the collateral damage:

(1) How many ISPs and ASes implement DNS injection-based censorship?

(2) How widely are DNS resolvers suffering from collateral damage due to censorship, and what is the cause of this collateral damage?

4.1 Searching for Injected Paths: Honey-Query

In order to measure the impact of injection on users outside the censored networks, we must first identify and exclude the networks which use DNS injection for censorship. Based on our previous experience with censored networks and the work of Lowe et al[7], we make two assumptions: the DNS injection occurs in the core or on the border of the networks and the DNS injector does not consider packet origin when injecting packets. If the censorship occurs in the edge connecting the user it is highly unlikely to cause collateral damage, and a censor which considers packet origin would not cause collateral damage.

Like the concept of a *Honeytoken* [11], we launch a large amount of *HoneyQueries* to search for the injected paths. These queries target non-responsive IPs with queries to a sensitive domain name. Because the query only targets non-nameservers, any DNS response is likely due to packet injection.

Probing Targets: In order to search all possible AS-level paths, ideally we should make sure that our HoneyQuery probing covers all ASes in the Internet. We select an IP address in each /24 of the IPv4 address and verify that the IPs are not running DNS servers. We then probe these 14 million target IPs with our HoneyQueries.

Vantage Point: Other observers [6, 7] and our own experience show that these injectors fake answers for both inbound and outbound DNS queries. Therefore, our HoneyQuery probing could possibly cover all ASes from a single vantage point as long as its not in a censored network. There does exist a minor false-positive: if an uncensored network receives transit from a censored network from our vantage point but not for other traffic. We are unable to determine when this occurs, and simply treat such networks as censored for later analysis. We selected a virtual private server (VPS) in AS 40676 (Psychz Networks) in US as our vantage point.

Domain Names For Testing: Experientially, we select 10 domain names for the probing(Table 1), including some social networks, pornography, web hosting, blogs, stream media, and search engines which we would expect to be targets of government or ISP censorship.

Domain Name	Category
`www.google.com`	Search Engines
`www.facebook.com`	Social Networks
`www.twitter.com`	Social Networks
`www.youbube.com`	Streaming Media
`www.yahoo.com`	News Portal
`www.appspot.com`	Web Hosting
`www.xxx.com`	Pornography
`www.urltrends.com`	Sites Ranking
`www.live.com`	Portal
`www.wikipedia.org`	Reference

Table 1: Domain Names for Probing.

Region	IP Count	Percentage
CN	388206	99.80
CA	363	0.09
US	127	0.03
HK	111	0.03
IN	94	0.02
Total 16 regions		

(a) Top 5 regions.

AS number	Region	IP Count	Percentage
4134	CN	140232	36.05
4837	CN	88573	22.77
4538	CN	35217	9.05
9394	CN	24880	6.40
4812	CN	14913	3.83
Total 197 ASes			

(b) Top 5 ASes.

Table 2: Statistics of the `Poisoned_IP_List` collected from HoneyQuery probing.

HoneyQuery Probing: We send HoneyQueries with domain names above to all the target IP addresses from the vantage point. If there is any response for a HoneyQuery, we mark the domain name as blacklisted and the target IP as a poisoned IP. We also collect all the IPs used in the injected responses(we call them lemon IPs). After HoneyQuery probing, we get three lists: (1)`Blacklisted_Domain_List`, containing poisoned domain names in testing domain name set; (2)`Poisoned_IP_List`, containing IPs suffering directly from censorship; (3)`Lemon_IP_List`, containing the IPs used in all the bogus responses (allowing us to recognize consistently censored results).

We conducted our HoneyQuery probing during November, 2001 and obtained a poisoned IP list of 388,988 IP addresses, distributed in 16 regions (CN, CA, US, HK, IN, AP, KR, JP, TW, DE, PK, AU, SG, ZA, SE, FI) and 197 ASes. The top regions and ASes are shown in Table 2.

For the IPs in the `Poisoned_IP_List`, its location (region or AS) does not mean that the hosting AS or region injects the faked DNS response; but means there should be an injector on the transit path from our vantage point. We will locate the injectors in § 4.2.

We obtained six domain names in the `Blacklisted_Domain_List`: `www.facebook.com, twitter.com, www.youtube.com, www.appspot.com, www.xxx.com, www.urltrends.com`, and 28 different IPs in the `Lemon_IP_List`, allowing us to easily

Region	Count	Percentage
US	12519	28.76
JP	4889	11.23
RU	3306	7.60
DE	2345	5.39
TW	1733	3.98
GB	1580	3.63
CA	1150	2.64
IT	1053	2.42
Total 173 regions		

Table 4: Distribution of open resolvers for StepNXQuery probing.

create queries that may experience censorship and a list of known-bad results

4.2 Locating Injecting ISPs: TraceQuery

Given the list of censored IPs, we now identify the network location of the injectors using a *TraceQuery*.

A TraceQuery is a crafted DNS query with a domain name in `Blacklisted_Domain_List` and a customized TTL in IP header. Like *traceroute*, TraceQuery utilizes TTL decrements to ensure that the packets expire in the network. When the query goes through the network, each router along the path will decrease the IP TTL by one. Once the IP TTL gets zero, the router will drop the packet, and send back an ICMP time exceed message, allowing us to record the network path. The queries which pass an injector also trigger a DNS reply before expiring.

By conducting a TraceQuery to the final destination in the Poisoned IP list, this reveals all the DNS injectors in the path and their locations in the network.

After TraceQuery probing, we obtained a list of 3,120 router IPs associated with DNS packet injection, belonging to only 39 Chinese ASes. Table 3 shows the information of top ten poisoning ASes. Thus we conclude that the non-Chinese IPs in our poisoned IP list are due to either errors in geolocation or Chinese transit for non-Chinese traffic.

4.3 Evaluating the Collateral Damage: StepNXQuery

Given the list of ASes that inject DNS replies, the question remains: does the censorship imposed within these ASes affect external resolvers? We probe for such collateral damage using a list of 43,842 non-censored open recursive resolvers distributed in 173 countries (Table 4).

We probe these resolvers from our non-poisoned vantage-point with names derived from the Blacklisted Domain List, comparing with the replies in the Lemon IP list to see if the resolver is generally poisoned. We conduct these probes using TCP, to further reduce the likelihood that the communication with the resolver encounters censorship.

Yet simple poisoning is not the only concern: if there exists a censored path from the resolver to the root, or from the resolver to a TLD authority, that path may also poison results. Thus we develop and utilize a series of *S*tepNXQueries. We structure these queries to take advantage of over-eager pattern matching in the censorship systems, which regard names such as `www.facebook.com.fu` as objectionable.

Thus we can guarantee that a query from the recursive resolver goes to a specific level in the DNS hierarchy by generating an NXNAME (No Such Name) triggering request. Thus, to test the root path from the resolver, we query for names like `www.facebook.com.{RANDOM}`, with `RANDOM` being a random string which will generate an NXNAME response from the root. By repeating this test 200 times with different random strings, we take advantage of the recursive resolver's willingness to distribute queries between authorities to test all paths to the root servers from the given resolver.

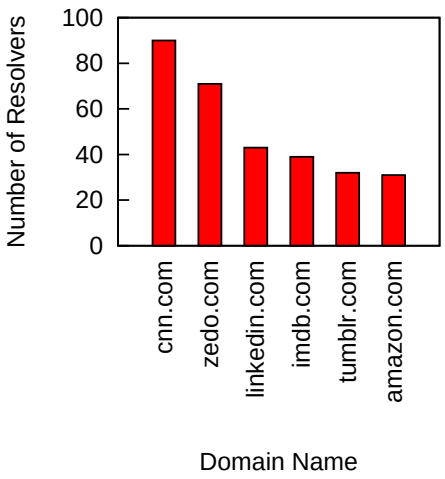


Figure 4: Affected domain names.

The same technique allows us to probe the path between the resolvers and the TLD servers, replacing {`RANDOM`} with {`RANDOM`}`.tld`. Since the TLD information is already cached with a long TTL, these queries only traverse the path between the resolver and the TLD authorities.

Finally, we find only 1 recursive resolver (124.219.23.209) in AS24154 in TW is poisoned because of collateral damage.

From the probing result, we can see that paths from recursive resolvers to root name servers seldom suffer from collateral damage, as the roots are heavily anycasted (except for the Chinese root servers), so DNS queries to the root seldom transit Chinese networks.

In contrast, the TLDs suffer from substantial collateral damage. We tested all of the 312 TLDs got from ICANN. For the three TLD in China (`.cn`, `.xn--fiqs8s`, `.xn--fiqz9s`), it is not a surprise that 43,322 (99.53%) resolvers return injected answers because the DNS resolution path have to get to the censored network.

Of greater concern is we find that 11,573 (26.40%) resolvers showed collateral damage for queries from one or more of 16 other TLDs. Figure 2 shows these TLDs and the number of affected resolvers. The second one, `.xn--3eb707e`, shares the same name infrastructure with the `.kr` ccTLD.

It seems strange that the number of affected resolvers for `.iq, .co, .travel, .no, .pl, .nz, .hk, .jp, .uk, .fi, .ca` are all around 90. We check the location of their name servers and find that it is not a coincidence: UltraDNS (AS12008) hosts the authority servers for all these TLDs.

Limited by space, we only present the detailed information for the most affected TLD: `.de`. As shown in Figure 3, over 70% of the resolvers from KR susceptible to collateral damage suffer collateral damage for `.de` queries, such as `www.epochtimes.de`.

Finally we constructed construct queries like KEYWORD-

AS Number	AS Name	Router IPs
4134	Chinanet	1952
4837	CNCGROUP China169 Backbone	489
4812	China Telecom (Group)	289
9394	CHINA RAILWAY Internet(CRNET)	78
9929	China Netcom Corp.	67
4808	CNCGROUP IP network China169 Beijing Province Network	55
9808	Guangdong Mobile Communication Co.Ltd.	38
17633	ASN for Shandong Provincial Net of CT	25
4538	China Education and Research Network Center	22
17816	China Unicom IP network China169 Guangdong province	19
Total 39 ASes		

Table 3: Information of top 10 injecting ASes.

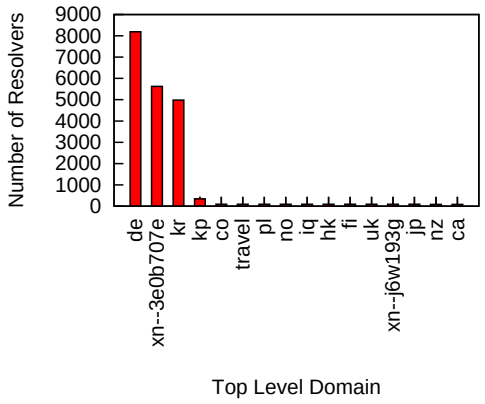
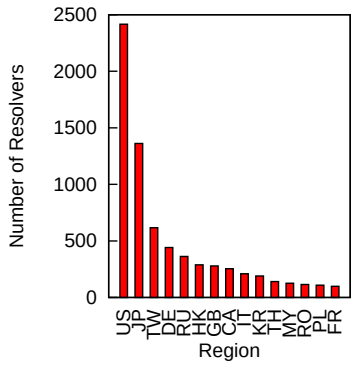
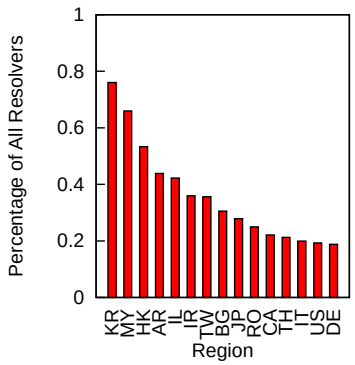


Figure 2: Affected TLD.



(a) Number of affected resolvers.



(b) Percentage of affected resolvers.

Figure 3: Distribution of affected resolvers for TLD `.de`.

.NXNAME.authority.tld (e.g., `www.twitter.com.abssdfds.ibm.com`) to explore paths from the resolvers to authoritative name servers for several domains.

We selected the top 82 popular domains from alexa.com, after excluding 18 Chinese sites. We see that queries for six domains could potentially trigger censorship on 30–90 resolvers, as shown in Figure 4. Although the number of affected domains and resolvers seem small comparing to the results of TLDs testing, this may represent the tip of the iceberg, considering the huge number of domain names of the whole Internet.

4.4 Further Analysis on Measurement Results

Table 5 gives the total number of resolvers suffering from collateral damage for root, TLDs and the top 82 domain names. 26.41% of experimental resolvers are polluted, distributed in 109 regions.

Unlike the worries presented by Mauricio [6], our measurement shows that the primary damage source arises from censored transit paths to TLD servers. According to Mauricio [6], the operator of I-Root server, Netnod, "withdrew their anycasted routes until their host (CNNIC) could secure assurances that the tampering would not recur". Our result partly confirmed their action. Since the roots themselves are highly anycasted, its unlikely that a path to a root needs to transit China.

In contrast, apparently a large amount of transit from the United States to Germany passes through China, resulting in the significant collateral damage to the `.de` ccTLD.

Rank	Region	Affected Resolver	Affected Rate
1	IR	157	88.20%
2	MY	163	85.34%
3	KR	198	79.20%
4	HK	403	74.63%
5	TW	1146	66.13%
6	IN	250	60.10%
10	IT	392	37.23%
14	JP	1437	29.39%
16	RU	835	25.26%
18	US	3032	24.22%
20	CA	272	23.65%
25	DE	470	20.04%
Total 109 Affected Regions			

Table 5: Collateral damage rate of different regions.

5. DISCUSSION

The cause of the collateral damage presented in this paper is the censorship activities by ISPs providing transit, not just connectivity. Although we'd hope otherwise, we believe it is naive to expect these ISPs to stop or avoid to applying DNS-injection based censorship activities, due to the significant social and political factors these ISPs face.

One possibility would be for the ISPs to apply more strict checks to avoid polluting transit queries. Although we do not support broad censorship activities, we hope that this

DNS Level	Affected Resolvers	Affected Rate
Root	1	0.002%
TLD	11573	26.40%
Authoritative	99	0.23%

Table 6: Number of affected resolvers in different level.

paper will raise awareness of the collateral damage caused by indiscriminate DNS censorship. If ISPs only act to censor customers, not transit, this prevents the collateral damage. However, because of the closed nature of the many censorship activities (such as the DNS filter in China), it is unclear to us if there are any technical challenges for those ISPs to implement such policy or not.

If the censoring ISPs do not change their current practice of DNS-injection, another possibility is for neighboring ISPs to consider them invalid for transit: the neighbors should prefer alternate paths and not advertise transit whenever an alternate path exists. In particular, the TLD operators should monitor their peering arrangements to check for censored paths.

Finally, and most importantly, DNSSEC naturally prevents this collateral damage, especially on the TLD level. Both the `.de` and `.kr` domains sign their results, enabling a DNSSEC-validating resolver which rejects the unsigned injected replies while waiting for the legitimate signed replies to avoid suffering collateral damage due to packet injection.

6. CONCLUSION

The contributions of this paper include:

(1) Comprehensive analysis of collateral damage by DNS injection. Iterative queries to different level of name servers, multiple name servers distributed in different locations and dynamic and anycast routing, are all factors which may cause a query to transit a censored network, even though both the user and the target are outside the censored area.

(2) Discovering and locating DNS injectors. We probed all the Internet to find the indiscriminate DNS injectors, locating these DNS injectors in 39 Chinese ASes.

(3) Measurement of affected recursive resolvers all over the world. We measured 43,842 open recursive resolvers in 173 countries, and found that 26.41% of them in 109 countries could be polluted.

(4) Primary path of pollution: from resolver to TLD servers. We find that the primary collateral damage arises from transit between the resolver and the TLD authorities, particularly the authorities for `.de` and `.kr`.

We expect to continue our study on the measurement of the collateral damage caused by DNS injection, using multiple vantage points and an expanded list of HoneyQueries. Although we has not come to a solution to allow recursive resolvers to be immune to the collateral damages from DNS-based censorship apart from DNSSEC validation, we hope our result can increase the Internet community's awareness of such behaviors, and take actions to actively detect and resist such pollution to the whole Internet.

7. REFERENCES

[1] M. A. Brown, D. Madory, A. Popescu, and E. Zmijewski. DNS Tampering and Root Servers, Nov. 2010. `http://www.renesys.com/tech/presentations/pdf/DNS-Tampering-and-Root-Servers.pdf`.

[2] R. Bush, M. Patton, R. Elz, and S. Bradner. Selection and Operation of Secondary DNS Servers. *RFC2182*, 1997.

[3] J. Crandall, D. Zinn, M. Byrd, E. Barr, and R. East. ConceptDoppler: A Weather Tracker for Internet Censorship. In *Proceedings of the 14th ACM Conference on Computer and Communications Security*, CCS'07, pages 352–365, New York, NY, USA, 2007. ACM.

[4] M. Dischinger, M. Marcon, S. Guha, K. P. Gummadi, R. Mahajan, and S. Saroiu. Glasnost: Enabling End Users to Detect Traffic Differentiation. In *Proceedings of the 7th USENIX conference on Networked systems design and implementation*, NSDI'10, pages 27–27, Berkeley, CA, USA, 2010. USENIX Association.

[5] R. Elz and R. Bush. Clarifications to the DNS Specification. *RFC2181*, 1997.

[6] M. V. Ereche. Odd Behaviour on One Node in I root-server, 2010. `https://lists.dns-oarc.net/pipermail/dns-operations/2010-March/005260.html`.

[7] G. Lowe, P. Winters, and M. L. Marcus. The Great DNS Wall of China. `http://cs.nyu.edu/~pcw216/work/nds/final.pdf`, 2007.

[8] P. Mockapetris. Domain Names - Concepts and Facilities. *RFC1034*, 1987.

[9] P. Mockapetris. Domain Names - Implementation and Specification. *RFC1035*, 1987.

[10] C. Partridge, T. Mendez, and W. Milliken. Host Anycasting Service. *RFC1546*, 1993.

[11] L. Spitzner. Honeytokens: The Other Honeypot. `http://www.symantec.com/connect/articles/honeytokens-other-honeypot`, 2003.

Mobile Data Charging: New Attacks and Countermeasures

Chunyi Peng Chi-yu Li Guan-hua Tu Songwu Lu Lixia Zhang

Department of Computer Science, University of California, Los Angeles, CA 90095
{chunyip, lichiyu, ghtu, slu, lixia}@cs.ucla.edu

ABSTRACT

3G/4G cellular networks adopt usage-based charging. Mobile users are billed based on the traffic volume when accessing data service. In this work, we assess both this metered accounting architecture and application-specific charging policies by operators from the security perspective. We have identified loopholes in both, and discovered two effective attacks exploiting the loopholes. The "toll-free-data-access-attack" enables the attacker to access any data service for free. The "stealth-spam-attack" incurs any large traffic volume to the victim, while the victim may not be even aware of such spam traffic. Our experiments on two operational 3G networks have confirmed the feasibility and simplicity of such attacks. We also propose defense remedies.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: General—*Security and protection*; C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*

Keywords

Cellular Networks, Mobile Data Services, Accounting Attacks

1. INTRODUCTION

Wireless access to Internet data services is getting increasingly popular, thanks to the deployment of 3G/4G cellular networks. Statistics from OECD [29] shows that, 62% broadband users in the US have subscribed to wireless data plans, with 137M subscribers in June 2010. There are also 1.2B mobile Web users worldwide already [28]. The explosive growth of smartphones (e.g., iPhones and Android phones) will further accelerate this usage trend.

While users enjoy wireless data access, it does not come for free. Most 3G/4G operators bill the user based on the usage data volume[1]. This metered charging[2] is officially stipulated by the 3G/4G standards. Based on the standards, charging is performed inside the cellular network (CN) on a per-flow basis. Each flow is defined by the five-tuple (source-IP, destination-IP, source-port, destination-port, protocol) or its subset. Whenever a data flow is initiated with the phone, the traffic volume is recorded at the CN when data traverse the CN to reach the phone/server. Therefore, the CN performs accounting operations based on its observed traffic volume. Carriers can also define their flow-specific billing policy.

In this paper, we present the first work that critically assesses the vulnerability of 3G/4G charging system. We discover loopholes in its policy practice and weakness in its charging architecture. As a result, we identify two new types of attacks against the charging system. In the *toll-free data access attack*, attackers can access any data service from the mobile phone for any period of time free of charge. It exploits the policy loopholes when the operator allows for certain free service (e.g., DNS service). In the *stealth spam attack*, attackers will inject arbitrarily large volume of spam data into the victim device, even after the target device has terminated its data service, thus fully unaware of such a spam. This attack exploits the architecture weakness of not using feedback from the user when making charging decisions, as well as features of Internet instant messaging applications (e.g., Skype and Google Talk). Our prototypes show that both attacks are feasible and simple enough to launch over the operational 3G networks. Our experiments on two 3G networks from two US carriers show that, the undercharge or overcharge traffic volume can go unbounded.

Three main contributions of this work are as follows: (1) We report the first security analysis on the 3G/4G network charging system and identify its loopholes; (2) We describe two new types of attacks, i.e., "toll-free-data-attack" and "stealth spam attack," which exploit the identified loopholes to undermine the charging system; we also use real experiments over two operational 3G networks to validate the feasibility and simplicity of these attacks and their potential damage; (3) We articulate the root cause for the existence of these loopholes and propose effective solutions to eliminating them. In summary, our study shows that, a dependable, metered charging system requires concerted coordination among the mobile device, the network, and applications. Security mechanisms are needed to strengthen every part and the overall system.

The rest of the paper is organized as follows. Section 2 introduces the 3G/4G architecture and its data charging system. Section

CCS'12, October 16–18, 2012, Raleigh, North Carolina, USA.

[1]In fact, operators do not offer unlimited monthly data plans for smartphone users any more. Both AT&T and Verizon effectively ended such plans for new customers in 2011, and T-mobile limits the high-speed data volume in its so-called unlimited data plan.

[2]In this work, we use the words "charging" and "accounting" interchangeably.

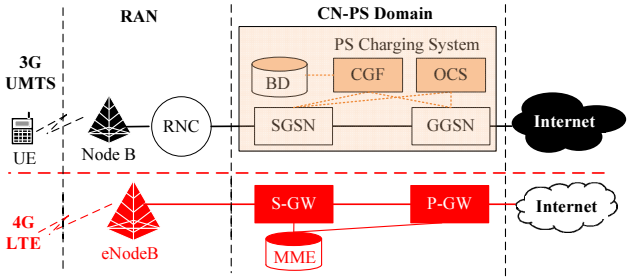


Figure 1: 3G/4G network architecture and charging components in PS domain.

3 analyzes the vulnerability of mobile data charging. Sections 4 and 5 describe the toll-free data access attack and stealth spam attack, as well as countermeasures, respectively. Section 6 compares with the related work, and Section 7 concludes the paper.

2. BACKGROUND

In this section, we introduce the charging architecture and process for mobile data services. Unlike the flat charging practice over the Internet, the current cellular network has been using usage-based charging for its data services. That is to say, the operator collects the actual usage volume over time for each user and imposes charges accordingly.

Broadly speaking, charging is performed on a per-connection basis. To communicate with a host on the Internet, the mobile device needs to first create a bearer service connection with the cellular network, which is further connected with the wired Internet. Once the connection is established, data packets are delivered. The connection has to traverse gateway-like devices (similar to routers in the Internet) in the cellular network core. These gateways then perform accounting operations by recording the data volume of those packets that traverse them, until the connection is completed.

Specifically, this accounting scheme has been shaped by both the standards and the operators' policy practice. The standards define the architecture and mechanisms, whereas the operators specify their own charging policies. Here, we mainly introduce the charging architecture and process in context of Universal Mobile Telecommunications System (UMTS), the most widely deployed 3G cellular network technology [19]. Note that, its charging mechanisms and operations are also applicable in 4G networks, e.g., Long Term Evolution (LTE) networks. For reference, Table 1 lists important acronyms used in this paper.

CDR	Charging Data Records
CN	Core Network
EH	External Host
FBC	Flow Based Charging
GGSN	Gateway GPRS Support Node
PDP	Packet Data Protocol
PS	Packet-Switched
SGSN	Serving GPRS Support Node
UE	User Equipment

Table 1: Table of important abbreviations and acronyms.

2.1 Data Charging Architecture

Figure 1 shows the overall 3G UMTS network architecture and charging system for data services. The UMTS network consists of the *Terrestrial Radio Access Network* (RAN) and the *core network* (CN). RAN provides wireless access to the mobile device (called User Equipment (UE)), and exchanges data session provisioning with the Packet-Switched (PS) core networks.

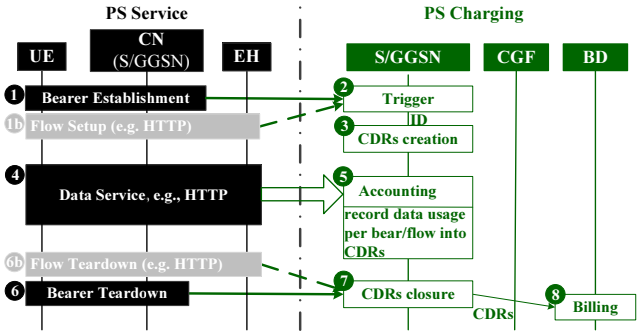


Figure 2: Charging procedures for a data service flow.

The major components of the PS core network are the *Serving GPRS Support Node* (SGSN) and the *Gateway GPRS Support Node* (GGSN). SGSN handles data packet delivery from and to the UEs within its geographical service area. GGSN acts as a router between the SGSN and the external wired Internet, and 'hides' the 3G UMTS infrastructure from the external network. In fact, SGSNs and GGSNs are the aforementioned gateway-like devices, recording data usage through them to perform charging functions.

Current cellular networks support both offline and online charging modes [17]. In addition to SGSN and GGSN, three more charging components work to support both modes: the *Billing Domain* (BD), the *Charging Gateway Function* (CGF), and the *Online Charging System* (OCS). In offline charging, data usage is collected during service provisioning in the form of *Charging Data Records* (CDRs), which are sent to the BD to generate data bills offline. The SGSN and GGSN are responsible for and generating CDRs. The CGF is used to validate CDRs from SGSNs/GGSNs and transfer CDRs to the BD. In online charging, mobile users have to pre-pay to obtain credits for data services in advance. The OCS authorizes whether or not users have enough credits so that GGSN/SGSN can proceed data services. GGSN/SGSN deducts data usage from the available credits and stops data services upon zero credit.

The charging subsystem for 4G LTE cellular network is almost identical to 3G UMTS. The major difference (also shown in Figure 1) is that, *Serving Gateway* (S-GW) and *Packet Data Network Gateway* (P-GW) [25] replace SGSN and GGSN to collect data usage and generate CDRs.

2.2 Data Charging Procedures

We next describe how mobile users are charged for data services through an example. Consider Alice is about to browse CNN news, thus starting a PS service (say, HTTP). Figure 2 illustrates the charging procedures (in the right) during the data service process (in the left), where the external host (EH) is `www.cnn.com`.

We first consider the offline charging mode. Initially, Alice has no available bearer service connection, which is used to carry one or multiple PS data services. She thus first establishes a bearer via Packet Data Protocol (PDP) Context Activation [15] (Step 1) where PDP contexts provide all the required information for IP packet data connections in cellular networks. Upon this activation, the UE is allowed to connect with the external data network through the SGSN and GGSN. This activation also triggers the charging procedure; the GGSN assigns a unique charging ID to the activated PDP context (Step 2). Using the charging ID, the SGSN and GGSN

start to create CDRs (Step 3), and are ready to record the upcoming data volume.

In addition to charging per bearer (PDP context), 3G operators also support charging per data flow, called as *Flow Based Charging* (FBC). FBC separates charging for different services (e.g., Web or VoIP) within the same PDP context [18]. The standard [16] specifies that one data flow is typically identified by five tuples: (1) source IP address, (2) source port number, (3) destination IP address, (4) destination port number, and (5) protocol ID of the protocol above IP, e.g., TCP or UDP. For example, a HTTP data flow can be represented by *(*, *, *, 80, TCP)*[3]. In this CNN case, FBC is triggered when Alice starts mobile Web browser and initiates a HTTP session (Step 1b).

Alice can read CNN news now. Both SGSN and GGSN route data packets between the UE and the external data network during the data service session (Step 4). In the meantime, the SGSN and GGSN record the traffic volume that *arrives* at them into corresponding CDRs (Step 5). They count the payload of GTP-U (*GPRS Tunneling Protocol- User Plane*) packets as data volume (see Figure 3); GTP-U delivers data within cellular networks and runs below the IP protocol.

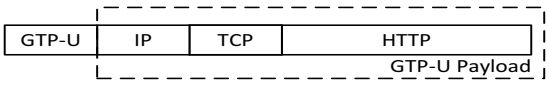


Figure 3: Example of of a GTP-U payload.

The accounting procedure (Step 5) lasts until this data service completes. It occurs when the UE tears down this bearer (Step 6) in bearer-based charging, or when Alice closes her HTTP session in flow-based charging (Step 6b). CDRs are subsequently closed and transferred to the BD (Step 7). Finally, the BD generates a billing item based on CDRs and assigns it to the proper user.

The online charging process is similar, though OCS participates in the triggering and accounting steps (Steps 2 and 5) by authenticating GGSN/SGSN to use user credits. There is also no need to send CDRs since the consumed credits are deducted during Step 5.

Regardless of the online/offline charging, the end goal is to ensure that the data usage recorded by the network is *indeed the same as the amount used (and wanted)* by a mobile device. The critical issue in mobile data charging is thus whether the accounting architecture and policy practice in cellular networks are secure enough to ensure proper billing. In this work, we seek to analyze such vulnerabilities, which malicious attackers can exploit to alter data usage record and make mobile users pay more (overcharging) or less (undercharging). We focus on the attack issues in offline charging, since the same issues also apply to online charging.

3. 3G ACCOUNTING VULNERABILITY

In this section, we provide an overview on the vulnerability issues we have identified in 3G data charging.

3.1 System Model

We focus on the security issues of the usage-based accounting rather than pricing, which sets the unit price for usage. In the typical scenario, a mobile user uses her/his smartphone to access the Internet data service via the 3G wireless network. Data communication is performed between the mobile phone and the Internet server/host. The study can be readily extended to the mobile to mobile communication setting.

As described in Section 2, the mobile user is charged for the data service (s)he uses. The operator records the data volume exchanged by the data service over time, and the user will pay based on the recorded usage volume. Specifically, the CN (more precisely, SGSN/GGSN) records the volume of the packets that traverse it and then charges the usage to the proper user via the mapping from the flow, the bearer to the corresponding UE and user, as illustrated in Figure 4.

[3]Each of the five tuples can be a wildcard.

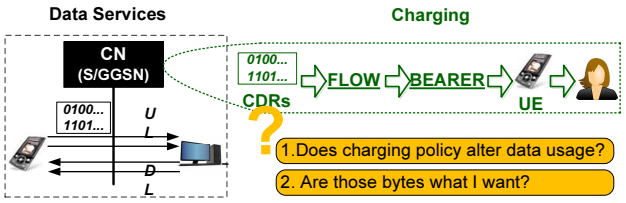


Figure 4: Issues in data charging practice.

We assume that the 3G charging subsystem is not compromised, i.e., all charging elements are operating properly (as described in Section 2). This implies that, the data usage records kept at SGSN/GGSN are not attacked, the mappings from CDRs to the flow, the bearer, and the mobile user are also intact. Moreover, user authentication within 3G/4G cellular networks works properly. Attackers cannot spoof other UE devices to access data services.

3.2 Two Achilles' Heels

Our study shows that 3G/4G accounting architecture and policy practice contain two loopholes, which can be exploited to launch charging-related attacks against the operator and mobile users.

The first relates to the charging policy that each 3G operator can define regarding what to be charged. Indeed, 3G/4G operators are allowed to adopt different charging policies. For example, they can charge Multimedia Messaging Service (MMS) service different from the common Internet data, or even provide free access for certain data services. The security implication is: Can the differential charging policy be exploited to alter the actual data usage? If two data services are charged differently, is it possible to fabricate the service type and masquerade as a cheaper one?

We study an extreme case of the above issue. Our findings in Section 4 show that, major US carriers usually offer free Domain Name Service (DNS) service, and all data usage associated with DNS service will be free. Our security question is: Given one type of free data service, is it possible to evade charges for other data services (e.g., standard Web browsing) as well? Our work confirms that it is indeed feasible. It is easy to exploit this loophole and launch an undercharging attack, during which the mobile user is charged for data volume smaller than its used amount, or even free of charge in the worst case. This attack defeats the fundamental principle of metered charging in all cellular networks.

The second loophole is rooted in the 3G/4G charging architecture. The core network records the packets, which traverse it and belong to specific flows, and charges the user accordingly. However, a question still remains: Is there any secure mechanism to verify with the user on whether the data would be indeed wanted by the user? What about those data bytes the attacker injects but mobile user never wants?

Our analysis in Section 5 shows that, current charging architecture lacks feedback mechanisms that allow the mobile user to explicitly express what packets are wanted or unwanted. Instead, operators decide on what packets are charged using their own rules. We further discover that, a mobile user is able to terminate the malicious (or suspicious) service on its application layer locally, but it cannot terminate the charging operations done at the carrier side.

Therefore, malicious attackers can inject spam packets and deceive the carriers to charge the mobile user for data volume larger than what it requests. Our experiments show that, the overcharging attack can be easily launched and there is no obvious upper bound on the overcharged volume. We note that, 3G/4G operators do provide security mechanisms via NAT and firewall. Consequently, a mobile device does not have a permanent and public IP address. It uses a private IP address and obtains temporary access to data services via NAT. The NAT-based operation ensures that the mobile user needs to initiate this service flow at the start. However, it is ineffective during the delivery process once the service flow starts. Consequently, it cannot shield incoming spam data when malicious hackers hijack the flow or when the victim later finds that (s)he is trapped.

3.3 Experimental Platform and Methodology

We design and conduct a series of experiments to examine security issues in 3G data charging. We now describe our platform and methods to obtain the data usage observed by the operator (V_{OP}) and the mobile phone (V_{UE}). The details of experiments are described in the followup sections.

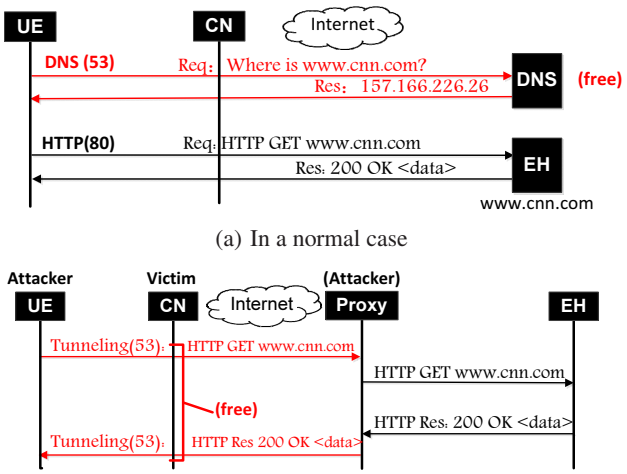
We run tests with two major mobile operators in the US, which together offer nationwide coverage for 102.3M users, thus claiming about 50% of US market. We denote them as Operator-I and Operator-II in this paper for privacy concerns. Our mobile devices use three Android phone models: HTC Desire, Samsung Galaxy S2, and Samsung Galaxy Note GT-N7000, running on Android 2.2, 2.3.4 and 2.3.6, respectively. Our experiments show that all the findings are phone platform independent. We use an ASUS EeeBox PC EB1501 desktop as the deployed host outside the cellular networks. It runs on an Intel Atom N330 1.6 GHz Dual Core processor and 1.5 GB DDR2 memory. This host acts as a content server (e.g., Web), proxy or attacker in various tests.

We use two methods to obtain data usage logged by operators. The first one is to dial a special number from the mobile phone to retrieve the remaining monthly data usage via a text message in a near real-time mode. Most operators support this Dial-In feature, e.g., via dialing #DATA for Verizon, *DATA# for AT&T, and #932# for T-Mobile in the US. By logging data usage before and after our experiment, we compute the usage volume observed by the operator during the experiment. The second method is to log onto the mobile carrier website and obtain itemized data usage records online. Based on the access availability, we choose the first method for Operator-I and the second for Operator-II. Both support 1 KB accuracy in their data usage report. Note that, data usage records only have timestamps. We use extra mechanisms to ensure that the usage record is exclusive to data services in our tests. We run factory reset first and disable "Background data" and "Auto-sync" features. We also use Wireshark [38], a monitoring tool to capture all-level packets to/from the phone to ensure clean environment.

To obtain data usage on mobile phones, we develop our own tool to use TrafficStats class interfaces [14] provided in Android SDK to collect network traffic statistics. We record the number of packets and bytes transmitted and received on all interfaces and on a per-application basis. We further use WireShark to log packet traces at our phones or deployed host if needed. We conduct each experiment for 5 –15 runs and average the results over these runs.

4. FREE MOBILE DATA ACCESS ATTACK

In this section, we report how attackers can obtain mobile data services for FREE. We find out that there exist loopholes in the current charging policy. Operators allow free data service for certain data flow, but do not enforce that the transmitted packets *indeed* belong to the designated free flow. Even worse, no effective mechanism is implemented to limit the traffic volume going through this free ride. Consequently, these loopholes can be exploited to enable any form of mobile data services for free. We use real experiments to examine security issues in the operators' charging practice, and describe three approaches to "free" data services. Finally, we make suggestions to fix this "bug."



(a) In a normal case

(b) Under a toll-free-data-attack

Figure 5: Web browsing in normal and attacked cases.

4.1 Loopholes in Charging Policy Practice

The 3G standards offer the operators flexibility to define their own charging policies. Unfortunately, their policies and implementations may contain serious flaws.

We use the example of Web browsing (`www.cnn.com`) to illustrate the vulnerabilities in the charging policy practice. Figure 5(a) illustrates typical steps for Web browsing. Upon receiving the target URL, the Web browser immediately initiates two actions. One is to send a DNS query to request the IP address for this URL. The other is to send a HTTP query to the Web server using the obtained IP address and receive a HTTP response. In mobile data charging, the above operations invoke two charging flows. One is the DNS query/response which goes through the CN to the DNS resolver or server. It is primarily carried by UDP on port 53, though TCP over port 53 is also allowed [32]. The other flow is for HTTP, which traverses the CN to enable communication between the UE and the Web server. It runs on TCP using port 80 (or other ports, e.g., 8080 or 443 for https). The CN records the data volume associated with each flow for billing.

Our study shows that, both operators tested in our experiments offer free DNS service. This policy practice makes sense, since DNS is considered a fundamental service for the Internet applications. Almost no Internet services can be initiated without DNS. DNS service is offered for free by many public DNS servers (e.g., Google, OpenDNS [4]). Operators thus have every reason not to charge DNS messages, to facilitate followup data usage by other Internet services. Therefore, free DNS service can be justified as a good (at least reasonable) policy.

However, our study shows that, there exist two loopholes to implement this free-DNS policy in reality. First, there is almost no enforcement mechanism to ensure that the packets going through this DNS-reserved port are indeed DNS messages (*free fake DNS loophole*). Second, there is no effective mechanism to limit the traf-

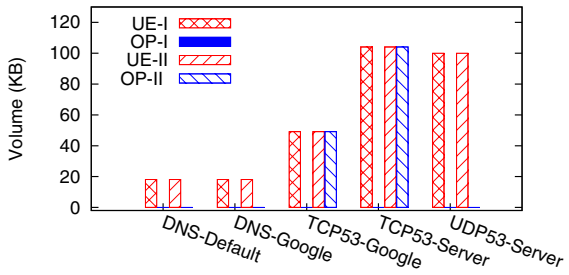


Figure 6: V_{UE} and V_{OP} in five DNS tests.

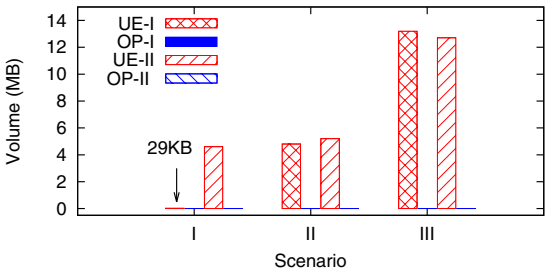


Figure 7: Feasibility test of free data services; $V_{OP} = 0$.

fic volume going through this port (*no volume-check loophole*). We next elaborate both using experiments.

• **Free fake DNS loophole** Our experiments show that, operators do not enforce free DNS service via the standard five-tuple flow ID (src IP, dest IP, src port, dest port, protocol). Instead, they use only the destination port (or plus protocol ID), thus exposing an obvious vulnerability.

We use experiments to verify whether the DNS service is free and what exact factors the free DNS service depends on in the operator's implementation. We conduct five experiments: (1) *DNS-Default*: the UE sends 100 DNS queries to the default DNS server provided by the operators; (2) *DNS-Google*: the UE sends the same DNS queries as (1) to a Google public DNS server (IP address: 8.8.8.8); (3) *TCP53-Google*: the UE sends the same DNS queries as (1) using TCP via port 53 to the above Google DNS server; (4) *TCP53-Server*: the UE sends 50 random packets to our own server using TCP via port 53, and require the server to return the received packets; each packet is 1KB, including IP/TCP headers; Source port number is randomly allocated; and (5) *UDP53-Server*: we repeat (4) but using UDP.

We conduct these experiments with two US major operators. We have purchased unlimited daily data plans from both operators and thus do not run into legal issues while testing free data services (the actual data usage is not counted by operators). We invalidate the hypothesis that the operator has no incentive to correctly report the traffic usage for users with unlimited access. To this end, we also use 200MB and 4GB data plans in free data service tests, and compare the results with using unlimited data plan. Results are consistent in all three plans. We further test different services (e.g., Web, YouTube, Gmail) using our unlimited data plans and verify that the data usage records at the UE and the operator are consistent. Figure 6 plots the data volume observed by the UE and two operators in all five cases. The results show that,

Operator-I: Packets via **port 53** are FREE Operator-II: Packets via **UDP + port 53** are FREE

Specifically, the UE sends and receives about 18.1 KB for 100 DNS queries and responses in both DNS-Default and DNS-Google tests. In the TCP53-Google test, the traffic volume rises to 48.1 KB due to TCP signaling overhead (SYNC, etc). In both TCP53-Server and UDP53-Server tests, the UE sends and receives 100 KB as expected. Operator-I charges for free (i.e., $V_{OP} = 0$) in all cases while Operator-II charges those TCP cases. From these results, we learn that the free DNS service is implemented by Operator-I using only one field in the flow ID (i.e., the destination port 53). In contrast, Operator-II enforces free DNS service using two tuples in the flow ID, i.e., UDP over destination port 53.

• **No volume-check loophole** Our study further shows that, there is no mechanism to limit the traffic volume going through this free-service port. To this end, we build our own server outside the cellular network that exchanges data services with mobile phones using UDP over port 53. We perform three experiments: (I) *Free-One*: the UE sends one request to our server to download a 5MB file; (II) *Free-Equal*: the UE uploads a 3MB file to our server, and requests to return the delivered packets; (III) *Free-Long*: the UE sends many small requests (100 B) to our server for an hour, each of which requests a 1KB response.

Figure 7 plots the data volume observed by the UE and both operators in the above three scenarios. It shows that, both operators can be exploited for free data services in all these scenarios, except that Operator-I does not allow unbounded traffic for one fake "DNS" request. The fake DNS message In the first test, Operator-I only allows to deliver 29 KB downlink data to the UE, while Operator-II delivers much larger file (up to 4 MB). We gauge that Operator-I might have enforced a checking mechanism to verify the size of the response message, in which a real DNS message size is typically bounded. However, this size checking can be easily bypassed. The UE simply sends out many small, dumb packets over this session, to increase the quota for downlink traffic. Then large downlink data can pass this checking. This has been validated in scenarios (II) and (III). In these tests, the gap between V_{UE} and the expected file size is mainly caused by unreliable transmission via UDP. These results demonstrate that free DNS service can be exploited to create any "free" data service.

4.2 Toll-Free Data Service Attack

We now show how to launch "free" mobile data access attack by using the above two loopholes. The key idea is to use a proxy server (placed outside the cellular network) to bridge the data access between the mobile phone and the Internet server. The communication between the proxy and the phone is carried out over the free channel (i.e., UDP or TCP over port 53, depending on the operator policy). We use "tunneling" between the UE and the proxy server. The proxy server relays packets on behalf of the UE. Free communication is thus extended to between the UE and an Internet host, while the 3G core network (CN) is the victim. Figure 5(b) illustrates the example of how Web browsing becomes free of charge. The process is similar to calling 800-voice hotlines, but for free data access. We thus name it as the "***toll-free-data-access-attack***."

We take three approaches when implementing the toll-free-data-attack. All show that, it is simple enough to obtain free mobile data access in reality. The first approach is to use a HTTP proxy running on port 53. It is easily done using available free proxy software such as FreeProxy [6]. The mobile Web browser is then configured to use the established HTTP proxy, as shown in Figure 8(a). This approach is easy to implement; no coding and hacking are needed. However, it only works for Web browsing and for Operator-I, which allows free TCP via port 53. To evaluate its effectiveness, we test two Web browsers – Mozilla Firefox and Opera Mobile [11], one

(a) HTTP proxy (b) Socks proxy (c) FDP proxy

Figure 8: Three approaches to "toll-free-data-access-attack."

hour each. We are able to use Operator-I network for free, while the actual data volume goes beyond 20 MB.

The second approach is to use a socks proxy. It works with various application protocols, e.g., HTTP, FTP, SMTP, POP3, NNTP, etc. Similarly, we deploy a socks proxy running on port 53. On the phone side, we install ProxyDroid [12] to enable socks proxy functionality. The phone configuration is shown in Figure 8(b). This method supports more applications without configuring each application individually. However, it still only applies to the TCP-53-free operators. We assess this attack with Operator-I using mobile applications, e.g., Web browsing, YouTube, Gmail, Google Map, Skype and FTP (via AndFTP [1]). The results show that, all services are free of charge except Skype voice call and FTP download. We figure out that, these two applications fail to go through the socks proxy; It is an implementation issue in ProxyDroid.

The third approach is to deploy a proxy server to enable "tunneling" between the phone and itself. To this end, we design a Free Data Protocol (FDP) to encapsulate data packets between the UE and the proxy into fake DNS messages, i.e., to carry packets in ANY-on-port-53 flows for Operator-I and UDP-on-port-53 flows for Operator-II. These messages are any data packets, not following DNS semantics. To bypass the limit of data volume for one fake DNS request (for Operator-I), FDP also periodically sends small KEEP-ALIVE messages from the UE side. The attacker enables the FDP at the UE and the proxy server. Note that, the DNS-tunneling idea is also used in the iodine [7] and NSTX [10] tools to enable Internet access over DNS. Moreover, the NSTX was used to demonstrates the similar idea for free Internet access with a toll-free Microsoft PPP dial-in number in Germany [8]. Both work in the wired Internet and free Internet access is available with specific DNS servers. In our experiments, we have built a simple prototype that revises applications to use FDP. We test our prototype with the revised HTTP and FTP applications working on top of FDP. Figure 8(c) captures the screen shot when visiting `www.cnn.com`. It shows that, data access is free for both operators while the actual data volume reaches 100 MB. Moreover, the upper limit of free traffic volume seems unbounded in our tests.

4.3 Suggestions to Fix the "Bug"

The simplest solution is to stop free DNS service or any other free data services that can go outside cellular networks. Fundamentally, for a metered charging service, people necessarily have incentives to exploit and abuse any transfer that is free. Therefore, the simplest, possibly also the best solution to abuse prevention is to eliminate the free services. Moreover, DNS traffic is negligible in normal cases; it should lead to no noticeable difference in most usage scenarios.

We also seek remedies to fix this bug while still retaining the free DNS service. For example, we have considered that the operator can provide quota for free DNS service. The DNS data usage beyond the quota will be still charged. Ideally, the quota should be assigned based on the average usage patterns. It can be a fixed amount or a percentage of the data usage. The challenge for this approach is how to set an appropriate quota. Some applications or services such as MobileMe [9] and DNSSEC [2] may heavily use DNS while others do not. The alternative approach is to enforce checking on the destination IP address of the DNS request. For example, free DNS services are only allowed when these messages go to designated or authenticated DNS resolvers or servers managed by carriers. However, it is still possible for attackers to deceive those resolvers/servers to forward fake "DNS" requests to a fake DNS server. The only difference is that the attack cost could be higher.

In the more general context, when the charging policy allows different unit-prices for diverse services (including free access to mobile Facebook [3] or a given Web site [5] in the extreme case), extra bullet-proof mechanisms are required; otherwise, the attacker always seeks to use the cheaper one. However, the deployment and operation of such security mechanisms will inevitably increase the cost of the carrier. Moreover, the security mechanism still needs to ensure itself to be secure in its design and operation. All these pose interesting research issues for the future.

5. STEALTH SPAM ATTACK

In this section, we describe the stealth spam attack, which is a new spam threat against mobile devices by exploiting the loopholes in current 3G/4G charging system. It stealthily injects a large volume of spam data, which the mobile device may not be even aware of (e.g., after the mobile device already closes the data session on its side). This incurs extra payment on the mobile user.

Stealth spam attack is different from conventional spam threats targeting mobile devices. Conventional spams include Email spam, SMS/MMS spam, junk image or video embedded in Web pages, etc. Users are typically aware of these annoying junk messages and may take actions to block them. In contrast, the stealth spam attack can be long lived, lasting several hours or more (observed in our experiments). The persistent spam session not only allows for the attacker to send a large volume of junk data, but also does it covertly. The users may be completely oblivious of such attacks.

5.1 Challenges and Opportunities

In practice, operators widely use NAT middlebox to handle IP address allocation of mobile devices [37]. Note that, attackers need to know the IP address of the phone when injecting spam data against the mobile device.

The deployment of NAT makes launching mobile spam attack a challenging task. Specifically, NAT offers two countermeasures against spam. First, it decouples network access from public reachability. The mobile UE is only allocated a private IP address (not reachable from the external network) when its bearer (i.e., PDP context) is activated. The UE is reachable from the public Internet only after NAT assigns it a translated IP address and a port number. This dynamic assignment only occurs when the UE initiates a data session (e.g., when starting a Google search or signing in Skype). Without the explicit activation from the UE side, data-charging operations never happen (as shown in the normal case of Figure 9). This tends to shield most conventional spam threats that send data to the UE via its IP address.

As the second countermeasure, operator's NAT boxes only grant temporary permissions for the traffic traversing the cellular core network. They only allow for the traffic to pass through within a provisional time window when the data session is alive. In the normal scenario, the charging time window ends when the UE terminates this data service. For example, mobile Web browser may immediately send a TCP FIN message to close the TCP connection, once the Web page is downloaded. This way, only within the

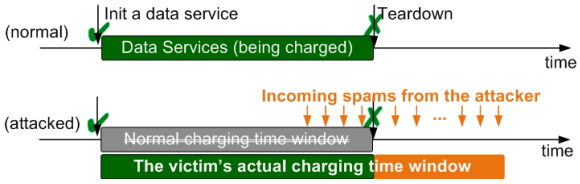


Figure 9: Illustration of stealth spam attack.

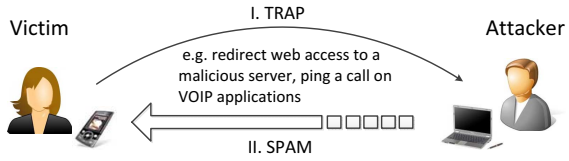


Figure 10: Steps to launch stealth spam attack.

given time window, those hosts, which know the access information (i.e., the translated IP address and the port number), are able to inject traffic to the UE. This window-controlled access also helps to protect the UE from spam threat. In addition, firewalls deployed by operators can also filter out spam.

On the other hand, the loopholes in the current 3G/4G charging system, as well as in applications, also create opportunities for stealth spam attack. Our analysis and experiments show that, there exist two loopholes in the current charging system. The first loophole is that,

> Data flow termination at the UE $\neq$ charging termination at the operator.

There exists inconsistency between the UE status and the operator's view on termination of a charging operation. When the user closes an application or an Internet service, (s)he thinks that the data flow is about to release and no more incoming traffic is allowed. However, the operator may view differently: This flow does not terminate as long as incoming packets belonging to this flow still arrive. The current 3G charging takes the operator's view. Therefore, charging can last much longer than expected. This occurs when the attacker starts this incoming spam before the normal teardown by the UE (shown in Figure 9)). We further find out that operators are unable to effectively stop data charging even when the UE explicitly sends teardown signals (e.g., in TCP). It is even worse for those UDP-based data service. The charging can last even longer once the spam starts; there is no sign for it to stop based on our experimental observation. We will elaborate them in next sections.

The second loophole is that,

> Initial authentication $\neq$ authentication during the whole data process.

All the authentication operations are performed at the start of the data flow (or when establishing the PDP context), but not when closing a flow. Therefore, the current charging procedure secures the initialization of the flow but not the whole process. Specifically, it cannot protect the data flow in the teardown process. The current design works for voice calls but not for data. Packet-switched IP data forwarding can push packets along different paths to reach the victim UE without prior consent, different from the circuit-switched fixed route for voice calls.

With these loopholes, stealth spam attack can be launched. Figure 10 shows two typical steps to launch this attack: trap and spam. First, it traps the UE to obtain its confidential access information

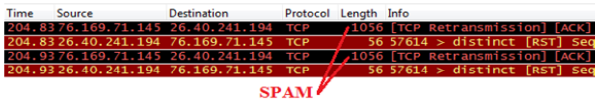


Figure 11: Wireshark traces at the victim even after the UE tear downs the TCP connection.

and flow permission to traverse the CN. The second step is to send junk packets. In the following, we describe how to implement them in several example scenarios and examine how badly it may hurt the victim.

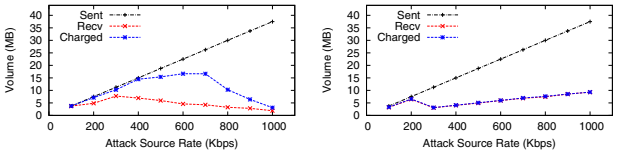
5.2 Spam Attack in TCP-based Services

We now describe how spam attack poses threats to those TCP-based services. Since TCP is a stateful procotol, we expect the spam to stop early once the UE application closes its TCP connection. Take Web browsing as an example. Once the Web page is fully retrieved, the Web browser may send a TCP FIN signal to the Web server and closes this TCP connection. Even though the Web server is malicious, the timeout mechanism also helps the UE to close this connection. The timer is typical set from tens of seconds to several minutes. However, our study has confirmed that the current charging practice contains loopholes. The operator may not stop charging, even when they can learn that the connection is closed by the UE.

In our experiments, we deploy a Web server as the attacker and modify its used TCP protocol. The spam attack starts when the UE clicks a malicious Web link and setups a TCP connection with the attacker. In the modified TCP, the normal TCP connection teardown procedure is disabled. This TCP will never send FIN or FIN-ACK signals like a normal TCP, upon receiving the teardown request from the UE. Once the UE is connected, the attacker immediately sends junk packets at a fixed rate for a given duration. To enable fixed-rate testing, we also disable TCP congestion control.

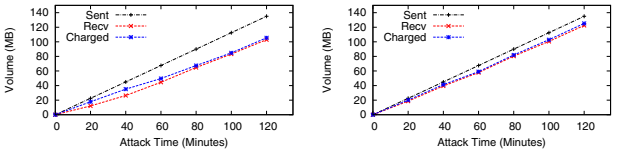
We first run experiments using various source rates for five minutes. Figure 12 plots the data volume increase due to this attack in both networks. It is observed that, as the incoming source rate grows beyond one threshold (about 400Kbps for Operator-I, 200Kbps for Operator-II), the attack seems to be blocked by the operator. The higher the source rate, the earlier the attack is blocked. For example, the spam is blocked in 24.7 seconds when the incoming rate reaches 1 Mbps for Operator-I while it gets blocked in 2 minutes for those attack at the source rates from 300 Kbps to 1 Mbps for Operator-II. This result implies that, operators do offer certain protection mechanism (e.g., blocking the TCP connection if it is too fast). However, these protection policies are operator specific. We find that, Operator-I may block the access to any data service while Operator-II only blocks this specific data service. We also observe that, the charging time window is not determined by the TCP connection status. When the UE closes this TCP connection, it sends TCP-RESET signals upon receiving spam packets. Figure 11 shows the Wireshark trace at the victim; TCP-RESET signals indicate that the UE aborts the connection and the 1056byte-packets are spam data units. The trace shows that, operators still allow the delivery of those spam packets and charge mobile users even when the UE TCP connection is closed.

We also test this attack at low source rate (150 Kbps) for various durations. Figure 13 plots the data volume increase in both operators. The low-rate attack can easily bypass the security check implemented by both operators. The attack can last for two hours; there is no sign to end during our experiments. The data volume incurred by this attack has exceeded 100 MB.



(a) Operator-I (b) Operator-II

Figure 12: Data volume caused by TCP-based stealth spam attacks under various source rates.



(a) Operator-I (b) Operator-II

Figure 13: Data volume caused by TCP-based stealth spam attacks for various durations.

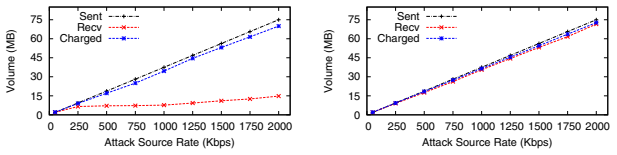
5.3 Spam Attack in UDP-based Services

We now describe stealth spam in UDP-based services. Since UDP is connectionless, it is even harder to decide when the UDP-based service ends and when the charging operation ends accordingly. The bad news is that, there is no clear protection mechanism for UDP-based service, while the operators at least use sort of abnormality-check for TCP-based sessions. The malicious attacker can launch stealth spam in UDP-based services by trapping the victim to open a UDP connection with itself. It may not be popular to use a malicious link to open UDP connection, we introduce to use two popular applications (e.g. VoIP and video streaming) to trap the victim and leak the access information.

• **Spam attack from your buddies** We demonstrate that the attacker can use VoIP service, including Skype and Google Talk, to construct the stealth spam. We use Skype as the example application. Skype is a globally used VoIP service and allows users to communicate with peers via voice, video, and instant messaging over the Internet [13]. Skype allows the buddies to communicate directly. A buddy on Skype has the chance to directly connect to the victim device without extra authentication.

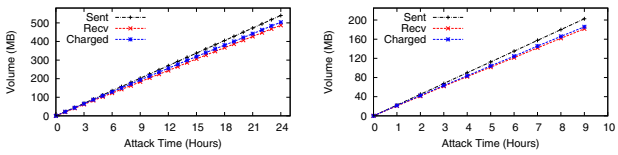
The first step to launch this attack is still to obtain the victim's confidential access information (i.e., translated IP address and port number) and its permission for this flow to traverse cellular networks (in the *trap* step). To this end, the attacker starts to make a call to this victim when he gets online using mobile phones. The attacker hangs up before the victim accepts the call, or even before the call rings at the victim side. This way, the victim may not be even unaware of this attempted call. During this process, the victim's Skype client performs two operations. First, it sends its access information to the attacker, which is proved in the attacker's Wireshark trace. In the meantime, it automatically notifies the operator that it accepts this flow, which subsequently grants the traffic flow from the spammer to traverse cellular networks. In the *spam* step, the attacker just sends junk UDP packets. The attacker can confirm that the victim is indeed a mobile user, based on the victim's translated IP address given by NAT. The operator-owned IP address block is readily known in advance. The spammer can also pick up the victim and launch operator-specific attacks.

We run experiments to validate this attack and verify whether extra checking mechanisms exist. We also vary the attack durations and incoming source rates in the tests. Figure 14 plots the overcharged volume versus different source rates during the five-



(a) Operator-I (b) Operator-II

Figure 14: Data volume caused by UDP-based (Skype) stealth spam attacks under various source rates.



(a) Operator-I (b) Operator-II

Figure 15: Data volume caused by UDP-based (Skype) stealth spam attacks for various durations.

minute Skype spam attack. The charging volume increase is in proportion to source rates. It implies that, operators do not enforce any security mechanism for UDP-based services. The spam volume can consequently grow much larger. We also make an interesting observation. In Operator-I, even though these packets may not be actually delivered to the UE (e.g., when the weak radio link cannot afford high-rate source), they are still charged by the operator. It shows that the operator might charge the mobile users based on the volume that arrive at them, not the one that they successfully delivery to the UE. Figure 15 plots the data volume caused by Skype stealth spam for various durations, with the source rate being 50kbps. It shows that the overcharge volume grows in proportion to the spam duration. There is no sign to end even when the attack has already lasted 24 hours for Operator-I (the overcharge volume reaches 500+ MB) during our experiments.

We also note that, the attack is still ongoing even after the victim signs out from Skype. The Wireshark trace at the victim side (see Figure 16) indicates that, spam packets still arrive at the UE and are charged by the operator after the UE logs out Skype. In the trace, the message of ICMP Port Unreachable shows that the UE has closed this application port after Skype logout.

In addition to Skype, this spam can be launched via Google Talk. The attacker also makes a call before the victim accepts it to trap the mobile user. The performance is similar; we omit it due to lack of space. Note that, the Skype/GTalk-based attack is a result of both 3G charging system vulnerability and Skype/GTalk implementation. The operator exposes the vulnerabilities at the first place, which still charges incoming spam packets that mobile application do not accept. The root cause is still that there is no feedback mechanism in the 3G charging system to tear down suspicious or malicious flows for mobile users. The Skype or other VoIP implementation (to release access information without explicit user confirmation) is exploited to mount this attack. Once you accept invitations from strangers or your buddies are compromised, you are vulnerable to this overcharging attack whenever you go online.

• **Spam attack in video streaming** Other channels exist to launch stealth spam attack in UDP-based services. Video streaming is another example. To trap the victim, the attacker can create a malicious link to redirect Web-browsing operations to start a realtime video streaming. For example, the victim may click one phishing link which redirects the victim's browser to:
rtsp://.*.1.204:554/trackID=5*,

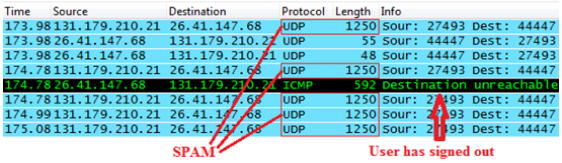


Figure 16: Wireshark traces at the victim after logout from Skype.

where RTSP is a network protocol to support video streaming [31]. Once the link is clicked, the victim automatically starts a new RTSP (over UDP) session running on port 554 and releases its confidential access information to the attacker. Once completed, the attacker blasts spam packets. We implement this attack and test it. We find that it performs similarly to Skype spam attack since both run on the top of UDP. We omit it due to lack of space. In both cases, UDP-based spam can inject an arbitrarily large volume of traffic and force the UE to pay more.

In summary, we have demonstrated that the stealthy spam attack is a real threat to mobile users. The attack is rooted in the inherent loopholes in the current charging architecture. Unless these loopholes are fixed, mobile users may always be victims when the stealth attack or more sophisticated attacks built on it are launched. On the other hand, the good news for mobile users is that, there is no obvious and strong incentive for attackers to launch such attacks now. Attackers cannot have immediate gains for themselves, unless an ill-intentioned operator contracts hackers to attack its own users for larger revenue gain or attack users in its competitor's network for unexpected user complaints, or a disgruntled attacker uses it to incur large monetary loss against his adversary. However, we quickly admit that incentive is an independent and interesting topic to study. Attackers may come up with unexpected incentives to launch more sophisticated attacks in this category in the future.

5.4 Remedy for Architecture Weakness

The fundamental problem underlying the stealth spam attack is that, there is no feedback mechanism from the UE to the carrier's charging system. So the operator cannot block unwelcome traffic based on the UE's feedback. This is an inherent design limitation in the current 3G/4G charging system. The IP-based push model makes spam attack easy. Any one can send to the UE without prior consent. Given the current architecture weakness, a viable charging system must have the following three components: (1) The mobile user himself must be aware of such potential attacks and apply precaution measures. He can simply limit the size of any automatic downloaded data (such as email fetching); (2) The UE must be able to detect unwanted traffic and send feedback. The current protocols at the network layer and the transport layer are designed with such feedback. However, many applications ignore unwanted data (e.g., Skype does so) in general. This has to be fixed to make them suitable to run over a metered charging service; (3) The carriers must take feedback from the UE to stop unwanted traffic.

Specifically, regarding the feedback mechanisms from the UE, we propose three solution options: *implicit-block*, *explicit-allow* and *explicit-stop*. The *implicit-block* solution is to enforce the CN components such as GGSN and NAT boxes. It uses implicit hints from the UE to justify whether the ongoing traffic is welcome or not. Once the traffic is unwanted, the CN blocks this flow and stops charging. The key issue is what messages can serve as hints on whether the UE's data packets are still wanted or not. For TCP-based service, TCP-RESET messages are sent from the UE if the the corresponding TCP connection is torn down earlier by the UE. Our study shows that, mobile Web browsers start to send TCP RESET messages upon receiving unintended TCP packets one-minute after they send the FIN signals. In case of UDP-based service, the UE responds a *ICMP Port Unreachable* message to the external sender upon receiving UDP packets on those closed ports. Therefore, messages of TCP-RESET and ICMP-Port-Unreachable can serve as the hints for the CN. To make correct decision, the CN can further exchange this information with the UE and seek confirmation from the mobile user. Using these implicit feedbacks, the CN should effectively disable the suspicious flows delivered to the UE. A downside of this solution is that, it takes effects only if the UE explicitly tear downs the service (e.g., quitting an application, terminating a TCP connection).

In the *explicit-allow* remedy, the UE explicitly specifies which packets are anticipated by adding/modifying the Packet Filters of Traffic Flow Template (TFT) associated with its PDP context. It can be done using *MS-Initiated PDP Context Modification Procedure* defined by 3GPP [15]. The attributes of the packet filter include [15]: (1) remote address; (2) local address; (3) protocol number, i.e., IPv4; (4) local port range; and (5) remote port range, etc. By adding packet filters, the UEs may not suffer from large spam attack when they are trapped or cheated to receive unexpected packets. One possible downside is that, it requires the UE to be fully aware of what it intends to send/receive. It requires detailed domain knowledge on various applications and services.

The *explicit-stop* solution is to provide explicit feedback from the UE to the carrier when closing some data services. Once the phone detects that there exists any malicious or suspicious flow, it immediately reports to the core network and asks to block such a flow. The spam flow can be detected by mobile anti-malware software, or identified by mobile applications or systems software (e.g., an exception is issued when the application layer or a lower layer in the protocol stack discards a large number of packets). Malicious attackers can also be detected through the collaboration of many phones [20]. This solution framework is flexible enough to integrate with different detection options. It also allows for the UE to stop data charging at any time, even when the UE was cheated or unaware of the attack at the start of the service. Its downside is that, current 3G/4G standards do not offer such mechanisms.

6. RELATED WORK

In recent years, security analysis on mobile devices has been an active research area (see [23, 24, 26, 36] for a few samples of the early work). Most of these studies focus on various types of mobile malware on various platforms of iOS, Android and Symbian, including virus [20, 26], spams such as SMS and making premium calls [23], DoS attacks [21, 27, 35], phishing [36], and privacy intrusion [24], etc. [33] has explored that unwanted traffic can cause large-scale wastage of logical resources in cellular networks. Our work uses real experiments to demonstrate that unwanted traffic can be cast to mobile victims and increase their payment. Certain types of these mobile malware such as viruses and SMS/MMS spams can also be used to incur overcharging attack as a byproduct. Despite these early efforts, security assessment of accounting system in the 3G/4G cellular networks remains a largely unaddressed topic. In this work, we provide the first experimental study that assesses the vulnerability, as well as new practical attacks, on the 3G/4G accounting system. We expose limitations in its charging architecture and loopholes in its policy practice. Both types of attacks described in this paper are also novel in 3G/4G security research.

Despite the popularity of 3G/4G data services, mobile data charging research (including pricing, accounting, billing) is still in its infancy. [22] provides a nice tutorial on pricing, charging, and billing methods for 3G systems up to 2005. [34] offers recent sur-

vey on pricing models, which are orthogonal to the accounting issue studied in this paper. [30] studies various cases of overcharging and undercharging in 3G networks but not from the security perspective. Finally, we note that several tools such as iodine, dns2tcp and NSTX [8] have been designed to circumvent data charging by wired Internet service providers. They are similar to our toll-free-data service approaches in principle, and we show that such ideas also work in wireless cellular networks.

7. CONCLUSION

The Internet is going wireless and mobile. Two driving forces for this trend have been the explosive growth of smartphones and the rapid deployment of 3G/4G infrastructure. Unlike the wired Internet, cellular networks have implemented usage-based charging, rather than the simpler flat-rate charging. The 3G/4G standards stipulate the accounting architecture, yet provide freedom for carriers to define their own charging policy. In this work, we conduct experiments on operational 3G networks to study the security implication of such an architecture and practice. We have discovered loopholes and showcased simple attacks, which are validated by experiments over two operational 3G networks.

Our study yields some insights. On the policy side, differential charging seems to be a popular practice for mobile data services. Given a metered charging system, people necessarily have incentives to exploit and abuse any transfer that is free. There is no simple, bullet-proof solution except eliminating the free service. In the more general problem setting, as long as differential charging exists among applications and services, attackers have incentives to abuse transfers that charge less. The free service simply exemplifies an extreme case. While the toll-free-data attack seems to be readily fixed, we believe that more fundamental issues need to be addressed in the long run. The current 3G/4G accounting architecture lacks proper validation and verification on the traversing traffic types and content, when offering differential charging for applications. The scalability of the associated security design also needs to be considered because of the increasing traffic diversity and volume, as well as the large user population. On the architecture side, the charging system records the data volume on behalf of users, but does not take any user feedback when making charging decisions. So the carrier cannot block unwelcome traffic by using feedback from users. The IP-based push model makes spam attack easier. Anyone can send to the UE without prior consent. Consequently, as confirmed by our experiments, victims may be charged for what they never anticipate, and attackers get data services they never pay.

Given the current architecture weakness, a dependable, usage-based charging system calls for concerted renovations among the network, the mobile device, and applications. The mobile user himself must be aware of such threats and apply precaution measures. The UE must be able to detect unwanted traffic and send feedback. Many applications lack such feedback mechanisms and simply ignore unwanted data, e.g., in the case of Skype. This must be fixed to make them suitable to run over a metered charging service. The operators must take feedback from the UE to stop unwanted traffic, and such feedback has to be carefully validated. The network also needs appropriate traffic validation and verification when making differential charging decisions for different applications and services. This work describes our current effort along this direction. We hope our preliminary study will stimulate further research on this important topic from both academia and industry.

Acknowledgment

We greatly appreciate the insightful comments and constructive feedback from the anonymous reviewers.

8. REFERENCES

[1] AndFTP - LYSESOFT, v2.9.8. http://www.lysesoft.com.
[2] DNSSEC. http://www.dnssec.net/.
[3] Fast and Free Facebook Mobile Access with 0.facebook.com. https://www.facebook.com/blog/blog.php?post=391295167130.
[4] Free Fast Public DNS Servers List. http://theos.in/windows-xp/free-fast-public-dns-server-list/.
[5] Free Gprs Mobile Tricks. http://darkwap.mobi/gprs-tricks/Free-Gprs-Mobile-Tricks.
[6] FreeProxy, v4.1. http://www.handcraftedsoftware.org/.
[7] Iodine. http://code.kryo.se/iodine/.
[8] IP Tunneling Through Nameservers. http://slashdot.org/story/00/09/10/2230242/ip-tunneling-through-nameservers.
[9] MobileMe. http://www.apple.com/mobileme/.
[10] NSTX. http://thomer.com/howtos/nstx.html.
[11] Opera Mobile, v12.0.0. http://www.opera.com/mobile/.
[12] ProxyDroid, v2.6.1. https://play.google.com/store/apps/details?id=org.proxydroid.
[13] Skype. http://www.skype.com.
[14] TrafficStats. http://developer.android.com.
[15] 3GPP. TS23.060: GPRS; Service description; Stage 2, Dec. 2006.
[16] 3GPP. TS23.125: Overall High Level Functionality and Architecture Impacts of Flow Based Charging, Mar 2006.
[17] 3GPP. TS32.240:Telecommunication management; Charging management; Charging architecture and principles, Sep. 2006.
[18] 3GPP. TS25.301: Radio Interface Protocol Architecture, 2008.
[19] G. America. Global 3G Deployments UMTS HSPA HSPA+, 2010.
[20] J. Cheng, S. H. Wong, H. Yang, and S. Lu. Smartsiren: virus detection and alert for smartphones. In *ACM MobiSys*, 2007.
[21] W. Enck, P. Traynor, P. McDaniel, and T. La Porta. Exploiting open functionality in sms-capable cellular networks. In *ACM CCS*, 2005.
[22] Z. Ezziane. Charging and Pricing Challenges for 3G systems. *IEEE Communications Surveys and Tutorials*, 7(1-4):58–68, 2005.
[23] A. P. Felt, M. Finifter, E. Chin, S. Hanna, and D. Wagner. A survey of mobile malware in the wild. In *Proceedings of SPSM'11*, 2011.
[24] C. Guo, H. Wang, and W. Zhu. Smartphone attacks and defenses. In *ACM HotNets-III*, 2004.
[25] H. Holma and A. Toskala. *LTE for UMTS: Evolution to LTE-Advanced*. Wiley, 2011.
[26] N. Leavitt. Mobile phones: The next frontier for hackers? *IEEE Computer*, 38(4):20–23, 2005.
[27] P. P. C. Lee, T. Bu, and T. Woo. On the Detection of Signaling DoS Attacks on 3G/WiMax Wireless Networks. *Computer Networks*, 53(15):2601–2616, Oct. 2009.
[28] mobiThinking. Global Mobile Statistics 2012. http://mobithinking.com/mobile-marketing-tools/latest-mobile-stats.
[29] OECD. Nearly Two-Thirds of US Broadband Subscribers are Wireless. http://www.websiteoptimization.com/bw/1012/.
[30] C. Peng, G. hua Tu, C. yu Li, and S. Lu. Can We Pay for What We Get in 3G Data Access? In *ACM MOBICOM*, 2012.
[31] RFC2326: Real Time Streaming Protocol (RTSP), 1998.
[32] RFC5966: DNS Transport over TCP - Implementation Requirements, 2010.
[33] F. Ricciato, P. Svoboda, E. Hasenleithner, and W. Fleischer. On the Impact of Unwanted Traffic onto a 3G Network. In *SecPerU'06*, 2006.
[34] S. Sen, C. Joe-Wong, S. Ha, and M. Chiang. Pricing Data: A Look at Past Proposals, Current Plans, and Future Trends. *CoRR*, abs/1201.4197, 2012.
[35] P. Traynor, M. Lin, M. Ongtang, V. Rao, T. Jaeger, P. McDaniel, and T. La Porta. On Cellular Botnets: Measuring the Impact of Malicious Devices on a Cellular Network Core. In *ACM CCS*, 2009.
[36] D. S. Wallach. Smartphone security: Trends and predictions. In *Secure Application Development*, SecAppDev, 2011.
[37] Z. Wang, Z. Qian, Q. Xu, Z. Mao, and M. Zhang. An Untold Story of Middleboxes in Cellular Networks. In *SIGCOMM*, 2011.
[38] WireShark. http://www.wireshark.org/.

‘‘Foiling the Cracker’’: A Survey of, and Improvements to, Password Security†

Daniel V. Klein

Software Engineering Institute
Carnegie Mellon University
Pittsburgh, PA 15217
dvk@sei.cmu.edu
+1 412 268 7791

ABSTRACT

With the rapid burgeoning of national and international networks, the question of system security has become one of growing importance. High speed inter-machine communication and even higher speed computational processors have made the threats of system ‘‘crackers,’’ data theft, data corruption very real. This paper outlines some of the problems of current password security by demonstrating the ease by which individual accounts may be broken. Various techniques used by crackers are outlined, and finally one solution to this point of system vulnerability, a proactive password checker, is proposed.

1. Introduction

The security of accounts and passwords has always been a concern for the developers and users of Unix. When Unix was younger, the password encryption algorithm was a simulation of the M-209 cipher machine used by the U.S. Army during World War II [Morris1979]. This was a fair encryption mechanism in that it was difficult to invert under the proper circumstances, but suffered in that it was too fast an algorithm. On a PDP-11/70, each encryption took approximately 1.25ms, so that it was possible to check roughly 800 passwords/second. Armed with a dictionary of 250,000 words, a cracker could compare their encryptions with those all stored in the password file in a little more than five minutes. Clearly, this was a security hole worth filling.

In later (post-1976) versions of Unix, the DES algorithm [DES1975] was used to encrypt passwords. The user’s password is used as the DES key, and the algorithm is used to encrypt a constant. The algorithm is iterated 25 times, with the result being an 11 character string plus a 2-character ‘‘salt.’’ This method is similarly difficult to decrypt (further complicated through the introduction of one of 4096 possible salt values) and had the added advantage of being slow. On a μVAX-II (a machine substantially faster than a PDP-11/70), a single encryption takes on the order of 280ms, so that a determined cracker can only check approximately 3.6 encryptions a second. Checking this same dictionary of 250,000 words would now take over 19 *hours* of CPU time. Although this is still not very much time to break a single account, there is no guarantee that this account will use one of these words as a password. Checking the passwords on a system with 50 accounts would take on average 40 CPU *days* (since the random selection of salt values practically guarantees that each user’s password will be encrypted with a different salt), with no guarantee of success. If this new, slow algorithm was combined with the user education needed to prevent the selection of obvious passwords, the problem seemed solved.

---

† This work was sponsored in part by the U.S. Department of Defense.

Regrettably, two recent developments and the recurrence of an old one have brought the problem of password security back to the fore.

1) CPU speeds have gotten increasingly faster since 1976, so much so that processors that are 25-40 times faster than the PDP-11/70 (e.g., the DECstation 3100 used in this research) are readily available as desktop workstations. With inter-networking, many sites have hundreds of the individual workstations connected together, and enterprising crackers are discovering that the ''divide and conquer'' algorithm can be extended to multiple processors, especially at night when those processors are not otherwise being used. Literally thousands of times the computational power of 10 years ago can be used to break passwords.

2) New implementations of the DES encryption algorithm have been developed, so that the time it takes to encrypt a password and compare the encryption against the value stored in the password file has dropped below the 1ms mark [Bishop1988, Feldmeier1989]. On a single workstation, the dictionary of 250,000 words can once again be cracked in under five minutes. By dividing the work across multiple workstations, the time required to encrypt these words against all 4096 salt values could be no more than an hour or so. With a recently described hardware implementation of the DES algorithm, the time for each encryption can be reduced to approximately 6 μs [Leong1991]. This means that this same dictionary can be be cracked in only 1.5 seconds.

3) Users are rarely, if ever, educated as to what are wise choices for passwords. If a password is in a dictionary, it is extremely vulnerable to being cracked, and users are simply not coached as to ''safe'' choices for passwords. Of those users who are so educated, many think that simply because their password is not in */usr/dict/words*, it is safe from detection. Many users also say that because they do not have any private files on-line, they are not concerned with the security of their account, little realizing that by providing an entry point to the system they allow damage to be wrought on their entire system by a malicious cracker.

Because the entirety of the password file is readable by all users, the encrypted passwords are vulnerable to cracking, both on-site and off-site. Many sites have responded to this threat with a reactive solution – they scan their own password files and advise those users whose passwords they are able to crack. The problem with this solution is that while the local site is testing its security, the password file is still vulnerable from the outside. The other problems, of course, are that the testing is very time consuming and only reports on those passwords it is able to crack. It does nothing to address user passwords which fall outside of the specific test cases (e.g., it is possible for a user to use as a password the letters ''qwerty'' – if this combination is not in the in-house test dictionary, it will not be detected, but there is nothing to stop an outside cracker from having a more sophisticated dictionary!).

Clearly, one solution to this is to either make */etc/passwd* unreadable, or to make the encrypted password portion of the file unreadable. Splitting the file into two pieces – a readable */etc/passwd* with all but the encrypted password present, and a ''shadow password'' file that is only readable by **root** is the solution proposed by Sun Microsystems (and others) that appears to be gaining popularity. It seems, however, that this solution will not reach the majority of non-Sun systems for quite a while, nor even, in fact, many Sun systems, due to many sites' reluctance to install new releases of software.†

What I propose, therefore, is a publicly available *proactive* password checker, which will enable users to change their passwords, and to check *a priori* whether the new password is ''safe.'' The criteria for safety should be tunable on a per-site basis, depending on the degree of security desired. For example, it should be possible to specify a minimum length password, a restriction that only lower case letters are not allowed, that a password that looks like a license plate be illegal, and so on. Because this proactive checker will deal with the pre-encrypted passwords, it will be able to perform more sophisticated pattern matching on the password, and will be able to test the safety without having to go through

---

† The problem of lack of password security is not just endemic to Unix. A recent Vax/VMS worm had great success by simply trying the username as the password. Even though the VMS user authorization file is inaccessible to ordinary users, the cracker simply tried a number of ''obvious'' password choices – and easily gained access.

the effort of cracking the encrypted version. Because the checking will be done automatically, the process of education can be transferred to the machine, which will instruct the user *why* a particular choice of password is bad.

2. Password Vulnerability

It has long been known that all a cracker need do to acquire access to a Unix machine is to follow two simple steps, namely:

1) Acquire a copy of that site's /*etc*/*passwd* file, either through an unprotected *uucp* link, well known holes in *sendmail*, or via *ftp* or *tftp*.

2) Apply the standard (or a sped-up) version of the password encryption algorithm to a collection of words, typically /*usr*/*dict*/*words* plus some permutations on account and user names, and compare the encrypted results to those found in the purloined /*etc*/*passwd* file.

If a match is found (and often at least one will be found), the cracker has access to the targeted machine. Certainly, this mode of attack has been known for some time [Spafford1988], and the defenses against this attack have also long been known. What is lacking from the literature is an accounting of just how vulnerable sites are to this mode of attack. In short, many people know that there is a problem, but few people believe it applies to them.

''There is a fine line between helping administrators protect their systems and providing a cookbook for bad guys.'' [Grampp1984] The problem here, therefore, is how to divulge useful information on the vulnerability of systems, without providing too much information, since almost certainly this information could be used by a cracker to break into some as-yet unviolated system. Most of the work that I did was of a general nature – I did not focus on a particular user or a particular system, and I did not use any personal information that might be at the disposal of a dedicated ''bad guy.'' Thus any results which I have been able to garner indicate only general trends in password usage, and cannot be used to great advantage when breaking into a particular system. This generality notwithstanding, I am sure that any self-respecting cracker would already have these techniques at their disposal, and so I am not bringing to light any great secret. Rather, I hope to provide a basis for protection for systems that can guard against future attempts at system invasion.

2.1. The Survey and Initial Results

In October and again in December of 1989, I asked a number of friends and acquaintances around the United States and Great Britain to participate in a survey. Essentially what I asked them to do was to mail me a copy of their /*etc*/*passwd* file, and I would try to crack their passwords (and as a side benefit, I would send them a report of the vulnerability of their system, although at no time would I reveal individual passwords nor even of their sites participation in this study). Not surprisingly, due to the sensitive nature of this type of disclosure, I only received a small fraction of the replies I hoped to get, but was nonetheless able to acquire a database of nearly 15,000 account entries. This, I hoped, would provide a representative cross section of the passwords used by users in the community.

Each of the account entries was tested by a number of intrusion strategies, which will be covered in greater detail in the following section. The possible passwords that were tried were based on the user's name or account number, taken from numerous dictionaries (including some containing foreign words, phrases, patterns of keys on the keyboard, and enumerations), and from permutations and combinations of words in those dictionaries. All in all, after nearly 12 CPU months of rather exhaustive testing, approximately 25% of the passwords had been guessed. So that you do not develop a false sense of security too early, I add that 21% (nearly 3,000 passwords) were guessed in the first week, and that in the first 15 minutes of testing, 368 passwords (or 2.7%) had been cracked using what experience has shown would be the most fruitful line of attack (i.e., using the user or account names as passwords). These statistics are frightening, and well they should be. On an average system with 50 accounts in the /*etc*/*passwd* file, one could expect the first account to be cracked in under 2 minutes, with 5–15 accounts being cracked by the end of the first day. Even though the **root** account may not be cracked, all it takes is one account being compromised for a cracker to establish a toehold in a system. Once that is done, any of a number of other well-known security loopholes (many of which have been

published on the network) can be used to access or destroy any information on the machine.

It should be noted that the results of this testing do not give us any indication as to what the *uncracked* passwords are. Rather, it only tells us what was essentially already known – that users are likely to use words that are familiar to them as their passwords [Riddle1989]. What new information it did provide, however, was the *degree* of vulnerability of the systems in question, as well as providing a basis for developing a proactive password changer – a system which pre-checks a password before it is entered into the system, to determine whether that password will be vulnerable to this type of attack. Passwords which can be derived from a dictionary are clearly a bad idea [Alvare1988], and users should be prevented from using them. Of course, as part of this censoring process, users should also be told *why* their proposed password is not good, and what a good class of password would be.

As to those passwords which remain unbroken, I can only conclude that these are much more secure and ‘‘safe’’ than those to be found in my dictionaries. One such class of passwords is word pairs, where a password consists of two short words, separated by a punctuation character. Even if only words of 3 to 5 lower case characters are considered, */usr/dict/words* provides 3000 words for pairing. When a single intermediary punctuation character is introduced, the sample size of 90,000,000 possible passwords is rather daunting. On a DECstation 3100, testing each of these passwords against that of a single user would require over 25 CPU *hours* – and even then, no guarantee exists that this is the type of password the user chose. Introducing one or two upper case characters into the password raises the search set size to such magnitude as to make cracking untenable.

Another ‘‘safe’’ password is one constructed from the initial letters of an easily remembered, but not too common phrase. For example, the phrase ‘‘Unix is a trademark of Bell Laboratories’’ could give rise to the password ‘‘UiatoBL.’’ This essentially creates a password which is a random string of upper and lower case letters. Exhaustively searching this list at 1000 tests per second with only 6 character passwords would take nearly 230 CPU days. Increasing the phrase size to 7 character passwords makes the testing time over 32 CPU *years* – a Herculean task that even the most dedicated cracker with huge computational resources would shy away from.

Thus, although I don’t know what passwords were chosen by those users I was unable to crack, I can say with some surety that it is doubtful that anyone else could crack them in a reasonable amount of time, either.

2.2. Method of Attack

A number of techniques were used on the accounts in order to determine if the passwords used for them were able to be compromised. To speed up testing, all passwords with the same salt value were grouped together. This way, one encryption per password per salt value could be performed, with multiple string comparisons to test for matches. Rather than considering 15,000 accounts, the problem was reduced to 4,000 salt values. The password tests were as follows:

1) Try using the user’s name, initials, account name, and other relevant personal information as a possible password. All in all, up to 130 different passwords were tried based on this information. For an account name **klone** with a user named ‘‘Daniel V. Klein,’’ some of the passwords that would be tried were: klone, klone0, klone1, klone123, dvk, dvkdvk, dklein, DKlein, leinad, nielk, dvklein, danielk, DvkkvD, DANIEL-KLEIN, (klone), KleinD, etc.

2) Try using words from various dictionaries. These included lists of men’s and women’s names (some 16,000 in all); places (including permutations so that ‘‘spain,’’ ‘‘spanish,’’ and ‘‘spaniard’’ would all be considered); names of famous people; cartoons and cartoon characters; titles, characters, and locations from films and science fiction stories; mythical creatures (garnered from Bulfinch’s mythology and dictionaries of mythical beasts); sports (including team names, nicknames, and specialized terms); numbers (both as numerals – ‘‘2001,’’ and written out – ‘‘twelve’’); strings of letters and numbers (‘‘a,’’ ‘‘aa,’’ ‘‘aaa,’’ ‘‘aaaa,’’ etc.); Chinese syllables (from the Pinyin Romanization of Chinese, a international standard system of writing Chinese on an English keyboard); the King James Bible; biological terms; common and vulgar phrases (such as ‘‘fuckyou,’’ ‘‘ibmsux,’’ and ‘‘deadhead’’);

keyboard patterns (such as ‘‘qwerty,’’ ‘‘asdf,’’ and ‘‘zxcvbn’’); abbreviations (such as ‘‘roygbiv’’ – the colors in the rainbow, and ‘‘ooottafagvah’’ – a mnemonic for remembering the 12 cranial nerves); machine names (acquired from */etc/hosts*); characters, plays, and locations from Shakespeare; common Yiddish words; the names of asteroids; and a collection of words from various technical papers I had previously published. All told, more than 60,000 separate words were considered per user (with any inter- and intra-dictionary duplicates being discarded).

3) Try various permutations on the words from step 2. This included making the first letter upper case or a control character, making the entire word upper case, reversing the word (with and without the aforementioned capitalization), changing the letter ‘o’ to the digit ‘0’ (so that the word ‘‘scholar’’ would also be checked as ‘‘sch0lar’’), changing the letter ‘l’ to the digit ‘1’ (so that ‘‘scholar’’ would also be checked as ‘‘scho1ar,’’ and also as ‘‘sch01ar’’), and performing similar manipulations to change the letter ‘z’ into the digit ‘2’, and the letter ‘s’ into the digit ‘5’. Another test was to make the word into a plural (irrespective of whether the word was actually a noun), with enough intelligence built in so that ‘‘dress’’ became ‘‘dresses,’’ ‘‘house’’ became ‘‘houses,’’ and ‘‘daisy’’ became ‘‘daisies.’’ We did not consider pluralization rules exhaustively, though, so that ‘‘datum’’ forgivably became ‘‘datums’’ (not ‘‘data’’), while ‘‘sphynx’’ became ‘‘sphynxs’’ (and not ‘‘sphynges’’). Similarly, the suffixes ‘‘-ed,’’ ‘‘-er,’’ and ‘‘-ing’’ were added to transform words like ‘‘phase’’ into ‘‘phased,’’ ‘‘phaser,’’ and ‘‘phasing.’’ These 14 to 17 additional tests per word added another 1,000,000 words to the list of possible passwords that were tested for each user.

4) Try various capitalization permutations on the words from step 2 that were not considered in step 3. This included all single letter capitalization permutations (so that ‘‘michael’’ would also be checked as ‘‘mIchael,’’ ‘‘miChael,’’ ‘‘micHael,’’ ‘‘michAel,’’ etc.), double letter capitalization permutations (‘‘MIchael,’’ ‘‘MiChael,’’ ‘‘MicHael,’’ ... , ‘‘mIChael,’’ ‘‘mIcHael,’’ etc.), triple letter permutations, and so on. The single letter permutations added roughly another 400,000 words to be checked per user, while the double letter permutations added another 1,500,000 words. Three letter permutations would have added at least another 3,000,000 words *per user* had there been enough time to complete the tests. Tests of 4, 5, and 6 letter permutations were deemed to be impracticable without much more computational horsepower to carry them out.

5) Try foreign language words on foreign users. The specific test that was performed was to try Chinese language passwords on users with Chinese names. The Pinyin Romanization of Chinese syllables was used, combining syllables together into one, two, and three syllable words. Because no tests were done to determine whether the words actually made sense, an exhaustive search was initiated. Since there are 398 Chinese syllables in the Pinyin system, there are 158,404 two syllable words, and slightly more than 16,000,000 three syllable words.† A similar mode of attack could as easily be used with English, using rules for building pronounceable nonsense words.

6) Try word pairs. The magnitude of an exhaustive test of this nature is staggering. To simplify this test, only words of 3 or 4 characters in length from */usr/dict/words* were used. Even so, the number of word pairs is $\mathbf{O}(10^7)$ (multiplied by 4096 possible salt values), and as of this writing, the test is only 10% complete.

For this study, I had access to four DECstation 3100’s, each of which was capable of checking approximately 750 passwords per second. Even with this total peak processing horsepower of 3,000 tests per second (some machines were only intermittently available), testing the $\mathbf{O}(10^{10})$ password/salt pairs for the first four tests required on the order of 12 CPU *months* of computations. The remaining two tests are still ongoing after an additional 18 CPU months of computation. Although for research purposes

———————

† The astute reader will notice that 398^3 is in fact 63,044,972. Since Unix passwords are truncated after 8 characters, however, the number of unique polysyllabic Chinese passwords is only around 16,000,000. Even this reduced set was too large to complete under the imposed time constraints.

this is well within acceptable ranges, it is a bit out of line for any but the most dedicated and resource-rich cracker.

2.3. Summary of Results

The problem with using passwords that are derived directly from obvious words is that when a user thinks ‘‘Hah, no one will guess this permutation,’’ they are almost invariably wrong. Who would ever suspect that I would find their passwords when they chose ‘‘fylgjas’’ (guardian creatures from Norse mythology), or the Chinese word for ‘‘hen-pecked husband’’? No matter what words or permutations thereon are chosen for a password, if they exist in some dictionary, they are susceptible to directed cracking. The following table give an overview of the types of passwords which were found through this research.

A note on the table is in order. The number of matches given from a particular dictionary is the total number of matches, irrespective of the permutations that a user may have applied to it. Thus, if the word ‘‘wombat’’ were a particularly popular password from the biology dictionary, the following table will not indicate whether it was entered as ‘‘wombat,’’ ‘‘Wombat,’’ ‘‘TABMOW,’’ ‘‘w0mbat,’’ or any of the other 71 possible differences that this research checked. In this way, detailed information can be divulged without providing much knowledge to potential ‘‘bad guys.’’

Additionally, in order to reduce the total search time that was needed for this research, the checking program eliminated both inter- and intra-dictionary duplicate words. The dictionaries are listed in the order tested, and the total size of the dictionary is given in addition to the number of words that were eliminated due to duplication. For example, the word ‘‘georgia’’ is both a female name and a place, and is only considered once. A password which is identified as being found in the common names dictionary might very well appear in other dictionaries. Additionally, although ‘‘duplicate,’’ ‘‘duplicated,’’ ‘‘duplicating’’ and ‘‘duplicative’’ are all distinct words, only the first eight characters of a password are used in Unix, so all but the first word are discarded as redundant.

Passwords cracked from a sample set of 13,797 accounts						
Type of Password	**Size of Dictionary**	**Duplicates Eliminated**	**Search Size**	**# of Matches**	**Pct. of Total**	**Cost/Benefit Ratio***
User/account name	130†	–	130	368	2.7%	2.830
Character sequences	866	0	866	22	0.2%	0.025
Numbers	450	23	427	9	0.1%	0.021
Chinese	398	6	392	56	0.4%‡	0.143
Place names	665	37	628	82	0.6%	0.131
Common names	2268	29	2239	548	4.0%	0.245
Female names	4955	675	4280	161	1.2%	0.038
Male names	3901	1035	2866	140	1.0%	0.049
Uncommon names	5559	604	4955	130	0.9%	0.026
Myths & legends	1357	111	1246	66	0.5%	0.053
Shakespearean	650	177	473	11	0.1%	0.023
Sports terms	247	9	238	32	0.2%	0.134
Science fiction	772	81	691	59	0.4%	0.085
Movies and actors	118	19	99	12	0.1%	0.121
Cartoons	133	41	92	9	0.1%	0.098
Famous people	509	219	290	55	0.4%	0.190
Phrases and patterns	998	65	933	253	1.8%	0.271
Surnames	160	127	33	9	0.1%	0.273
Biology	59	1	58	1	0.0%	0.017
/usr/dict/words	24474	4791	19683	1027	7.4%	0.052
Machine names	12983	3965	9018	132	1.0%	0.015
Mnemonics	14	0	14	2	0.0%	0.143
King James bible	13062	5537	7525	83	0.6%	0.011
Miscellaneous words	8146	4934	3212	54	0.4%	0.017
Yiddish words	69	13	56	0	0.0%	0.000
Asteroids	3459	1052	2407	19	0.1%	0.007
Total	86280	23553	62727	**3340**	**24.2%**	0.053

The results are quite disheartening. The total size of the dictionary was only 62,727 words (not counting various permutations). This is much smaller than the 250,000 word dictionary postulated at the beginning of this paper, yet armed even with this small dictionary, nearly 25% of the passwords were cracked!

---

* In all cases, the cost/benefit ratio is the number of matches divided by the search size. The more words that needed to be tested for a match, the lower the cost/benefit ratio.

† The dictionary used for user/account name checks naturally changed for each user. Up to 130 different permutations were tried for each.

‡ While monosyllablic Chinese passwords were tried for all users (with 12 matches), polysyllabic Chinese passwords were tried only for users with Chinese names. The percentage of matches for this subset of users is 8% – a greater hit ratio than any other method. Because the dictionary size is over 16×10^6, though, the cost/benefit ratio is infinitesimal.

Length of Cracked Passwords		
Length	**Count**	**Percentage**
1 character	4	0.1%
2 characters	5	0.2%
3 characters	66	2.0%
4 characters	188	5.7%
5 characters	317	9.5%
6 characters	1160	34.7%
7 characters	813	24.4%
8 characters	780	23.4%

The results of the word-pair tests are not included in either of the two tables. However, at the time of this writing, the test was approximately 10% completed, having found an additional 0.4% of the passwords in the sample set. It is probably reasonable to guess that a total of 4% of the passwords would be cracked by using word pairs.

3. Action, Reaction, and Proaction

What then, are we to do with the results presented in this paper? Clearly, something needs to be done to safeguard the security of our systems from attack. It was with intention of enhancing security that this study was undertaken. By knowing what kind of passwords users use, we are able to prevent them from using those that are easily guessable (and thus thwart the cracker).

One approach to eliminating easy-to-guess passwords is to periodically run a password checker – a program which scans */etc/passwd* and tries to break the passwords in it [Raleigh1988]. This approach has two major drawbacks. The first is that the checking is very time consuming. Even a system with only 100 accounts can take over a month to diligently check. A halfhearted check is almost as bad as no check at all, since users will find it easy to circumvent the easy checks and still have vulnerable passwords. The second drawback is that it is very resource consuming. The machine which is being used for password checking is not likely to be very useful for much else, since a fast password checker is also extremely CPU intensive.

Another popular approach to eradicating easy-to-guess passwords is to force users to change their passwords with some frequency. In theory, while this does not actually eliminate any easy-to-guess passwords, it prevents the cracker from dissecting */etc/passwd* ‘‘at leisure,’’ since once an account is broken, it is likely that that account will have had it’s password changed. This is of course, only theory. The biggest disadvantage is that there is usually nothing to prevent a user from changing their password from ‘‘Daniel’’ to ‘‘Victor’’ to ‘‘Klein’’ and back again (to use myself as an example) each time the system demands a new password. Experience has shown that even when this type of password cycling is precluded, users are easily able to circumvent simple tests by using easily remembered (and easily guessed) passwords such as ‘‘dvkJanuary,’’ ‘‘dvkFebruary,’’ etc [Reid1989]. A good password is one that is easily remembered, yet difficult to guess. When confronted with a choice between remembering a password or creating one that is hard to guess, users will almost always opt for the easy way out, and throw security to the wind.

Which brings us to the third popular option, namely that of assigned passwords. These are often words from a dictionary, pronounceable nonsense words, or random strings of characters. The problems here are numerous and manifest. Words from a dictionary are easily guessed, as we have seen. Pronounceable nonsense words (such as ‘‘trobacar’’ or ‘‘myclepate’’) are often difficult to remember, and random strings of characters (such as ‘‘h3rT+aQz’’) are even harder to commit to memory. Because these passwords have no personal mnemonic association to the users, they will often write them down to aid in their recollection. This immediately discards any security that might exist, because now the password is visibly associated with the system in question. It is akin to leaving the key under the door mat, or writing the combination to a safe behind the picture that hides it.

A fourth method is the use of ‘‘smart cards.’’ These credit card sized devices contain some form of encryption firmware which will ‘‘respond’’ to an electronic ‘‘challenge’’ issued by the system onto which the user is attempting to gain acccess. Without the smart card, the user (or cracker) is unable to

respond to the challenge, and is denied access to the system. The problems with smart cards have nothing to do with security, for in fact they are very good warders for your system. The drawbacks are that they can be expensive and must be carried at all times that access to the system is desired. They are also a bit of overkill for research or educational systems, or systems with a high degree of user turnover.

Clearly, then, since all of these systems have drawbacks in some environments, an additional way must be found to aid in password security.

3.1. A Proactive Password Checker

The best solution to the problem of having easily guessed passwords on a system is to prevent them from getting on the system in the first place. If a program such as a password checker *reacts* by detecting guessable passwords already in place, then although the security hole is found, the hole existed for as long as it took the program to detect it (and for the user to again change the password). If, however, the program which changes user's passwords (i.e., /*bin*/*passwd*) checks for the safety and guessability *before* that password is associated with the user's account, then the security hole is never put in place.

In an ideal world, the proactive password changer would require eight character passwords which are not in any dictionary, with at least one control character or punctuation character, and mixed upper and lower case letters. Such a degree of security (and of accompanying inconvenience to the users) might be too much for some sites, though. Therefore, the proactive checker should be tuneable on a per-site basis. This tuning could be accomplished either through recompilation of the *passwd* program, or more preferably, through a site configuration file.

As distributed, the behavior of the proactive checker should be that of attaining maximum password security – with the system administrator being able to turn off certain checks. It would be desireable to be able to test for and reject all password permutations that were detected in this research (and others), including:

- Passwords based on the user's account name
- Passwords based on the user's initials or given name
- Passwords which exactly match a word in a dictionary (not just /*usr*/*dict*/*words*)
- Passwords which match a word in the dictionary with some or all letters capitalized
- Passwords which match a reversed word in the dictionary
- Passwords which match a reversed word in the dictionary with some or all letters capitalized
- Passwords which match a word in a dictionary with an arbitrary letter turned into a control character
- Passwords which match a dictionary word with the numbers '0', '1', '2', and '5' substituted for the letters 'o', 'l',
- Passwords which are simple conjugations of a dictionary word (i.e., plurals, adding ''ing'' or ''ed'' to the end of the word, etc.)
- Passwords which are patterns from the keyboard (i.e., ''aaaaaa'' or ''qwerty'')
- Passwords which are shorter than a specific length (i.e., nothing shorter than six characters)
- Passwords which consist solely of numeric characters (i.e., Social Security numbers, telephone numbers, house addresses or office numbers)
- Passwords which do not contain mixed upper and lower case, or mixed letters and numbers, or mixed letters and punctuation
- Passwords which look like a state-issued license plate number

The configuration file which specifies the level of checking need not be readable by users. In fact, making this file unreadable by users (and by potential crackers) enhances system security by hiding a valuable guide to what passwords *are* acceptable (and conversely, which kind of passwords simply cannot be found).

Of course, to make this proactive checker more effective, it woule be necessary to provide the dictionaries that were used in this research (perhaps augmented on a per-site basis). Even more importantly, in addition to rejecting passwords which could be easily guessed, the proactive password changer would also have to tell the user *why* a particular password was unacceptable, and give the user suggestions as to what an acceptable password looks like.

4. Conclusion (and Sermon)

It has often been said that ‘‘good fences make good neighbors.’’ On a Unix system, many users also say that ‘‘I don’t care who reads my files, so I don’t need a good password.’’ Regrettably, leaving an account vulnerable to attack is not the same thing as leaving files unprotected. In the latter case, all that is at risk is the data contained in the unprotected files, while in the former, the whole system is at risk. Leaving the front door to your house open, or even putting a flimsy lock on it, is an invitation to the unfortunately ubiquitous people with poor morals. The same holds true for an account that is vulnerable to attack by password cracking techniques.

While it may not be actually true that good fences make good neighbors, a good fence at least helps keep out the bad neighbors. Good passwords are equivalent to those good fences, and a proactive checker is one way to ensure that those fences are in place *before* a breakin problem occurs.

References

Morris1979.
Robert T. Morris and Ken Thompson, ‘‘Password Security: A Case History,’’ *Communications of the ACM*, vol. 22, no. 11, pp. 594-597, November 1979.

DES1975.
‘‘Proposed Federal Information Processing Data Encryption Standard,’’ *Federal Register (40FR12134)*, March 17, 1975.

Bishop1988.
Matt Bishop, ‘‘An Application of a Fast Data Encryption Standard Implementation,’’ *Computing Systems*, vol. 1, no. 3, pp. 221-254, Summer 1988.

Feldmeier1989.
David C. Feldmeier and Philip R. Karn, ‘‘UNIX Password Security – Ten Years Later,’’ *CRYPTO Proceedings*, Summer 1989.

Leong1991.
Philip Leong and Chris Tham, ‘‘UNIX Password Encryption Considered Insecure,’’ *USENIX Winter Conference Proceedings*, January 1991.

Spafford1988.
Eugene H. Spafford, ‘‘The Internet Worm Program: An Analysis,’’ Purdue Technical Report CSD-TR-823, Purdue University, November 29, 1988.

Grampp1984.
F. Grampp and R. Morris, ‘‘Unix Operating System Security,’’ *AT&T Bell Labs Technical Journal*, vol. 63, no. 8, pp. 1649-1672, October 1984.

Riddle1989.
Bruce L. Riddle, Murray S. Miron, and Judith A. Semo, ‘‘Passwords in Use in a University Timesharing Environment,’’ *Computers & Security*, vol. 8, no. 7, pp. 569-579, November 1989.

Alvare1988.
Ana Marie De Alvare and E. Eugene Schultz, Jr., ‘‘A Framework for Password Selection,’’ *USENIX UNIX Security Workshop Proceedings*, August 1988.

Raleigh1988.
T. Raleigh and R. Underwood, ‘‘CRACK: A Distributed Password Advisor,’’ *USENIX UNIX Security Workshop Proceedings*, August 1988.

Reid1989.
Dr. Brian K Reid, DEC Western Research Laboratory, 1989. Personal communication.

QryGraph: A Graphical Tool for Big Data Analytics

Sanny Schmid, Ilias Gerostathopoulos, Christian Prehofer
Fakultät für Informatik
Technische Universität München
Munich, Germany
{schmidsa, gerostat, prehofer}@in.tum.de

***Abstract*—The advent of Big Data has created a rich set of diverse languages and tools for data manipulation and analytics within the Hadoop ecosystem. Pig has a prominent role within this ecosystem as a scripting layer—a convenient way to create analytics jobs that are issued for batch processing in a Hadoop cluster. In order to leverage the benefits of graphical domain specific languages, namely intuitive visual design and inspection, we implemented a web-based graphical tool called QryGraph that complements Pig in various ways. First, it allows a user to create Pig queries in a graphical editor and check their syntax. Second, it provides an administrative interface for managing the execution and overall lifecycle of Pig queries. Finally, it will allow for debugging by running queries on test data sets and for creating user-defined query sub-graphs that can be reused across different Pig queries.**

Keywords—Big Data; tool support; Pig language

I. INTRODUCTION

Recent advances in Big Data and Internet of Things technologies have created new disruptive possibilities related to new insight that can be obtained by analyzing the large quantities of sensed data [1]–[3]. More and more enterprises are looking into ways to build value-added services on top of Big Data analytics infrastructures.

In this context, Apache Hadoop has emerged as the de facto standard in Big Data analytics. Hadoop is an open-source ecosystem comprised of a multitude of languages and tools for data manipulation and analytics, e.g. MapReduce, HDFS, Hive, Pig, HBase, Kafka, Storm, Spark, etc. It supports processing of both static data—batch mode—and streams of incoming data in a real-time fashion—stream mode. At the same time, it comprises tools to address the needs of both developers, administrators, and data scientists and analysts. The main advantage of Hadoop is that it provides a fault-tolerant infrastructure that can easily scale to several thousand nodes in a single cluster and to several petabytes of data. The data stored in a Hadoop cluster are analyzed in batch mode by developing application-specific "mappers" and "reducers", i.e. functions that work on key/value pairs: mappers operate on input data (e.g. a large file residing in the Hadoop Distributed File System—HDFS [4]); reducers combine and summarize the results of the mappers [5]. Once the application-specific mapper and the reducer is implemented in Java or another Hadoop-compliant implementation language, they are bundled together into a single analytics job (also called a *map-reduce program*) that is issued to the Hadoop cluster. This triggers the parallel execution of several mapper and reducer tasks; the end result is then written back to the HDFS.

Within the Hadoop ecosystem, the *Pig* platform and language (also called Pig Latin) provide a convenient way to create analytics jobs compared to the manual implementation of mappers and reducers in a general-purpose programming language such as Java [6], [7]. Being a domain-specific language, Pig is essentially a thin layer over Hadoop that allows for specifying succinct scripts for analytics jobs that load data, apply transformations on the data, and store the final results.

Pig has been significantly enhanced since its inception at Yahoo! Research in 2006 to include several advanced features such as error handling and type checking, together with several performance improvements [7]. We nevertheless believe that there is still room for improvement, in particular in supporting the users of Pig (typically developers) in the creation, validation and evolution of complex analytics jobs.

To this end, we implemented a tool for Big Data analytics called QryGraph. Our tool supports Pig users in creating new analytics jobs that comply with the Pig language via an intuitive graphical editor. The editor allows for both visual design and validation via visual inspection. It provides an administrative interface for managing the execution and overall lifecycle of analytics jobs, including testing and debugging features. Finally, it will provide enhanced debugging features, such as running queries on test data sets, and will allow creating user-defined query sub-graphs that can be reused across different Pig queries.

In this artifact paper, we present the main functionalities and design goals of QryGraph, together with its technical architecture and extensibility mechanisms. We detail on the ongoing implementation and articulate our long-term plans. Our initial experience with using the tool indicates that it could become a vehicle for enhancing the understanding, creation, and evolution of Big Data analytics.

The rest of the text is structured as follows. Section II presents the running example and its solution in Pig. Section III presents our tool and discusses the main features and usage scenarios. Section IV reflects on our experience so far with QryGraph, and its current limitations. Finally, Section V compares our approach in supporting Big Data analytics to the state of the art and practice, and Section VI concludes and presents our future work plan.

978-1-5090-1897-0/16/$31.00 ©2016 IEEE

car_id	speed	longitude	latitude	timestamp
1	20	48.137	11.577	55
2	7	48.141	11.532	66
3	0	48.187	11.521	77

PL_id	available	longitude	latitude	timestamp
1	1	48.198	11.578	31
2	1	48.126	11.512	44
3	0	48.132	11.501	55

Fig. 1. Exemplary data sets in the running example.

II. Running Example and Background

A. Running Example

In our running example, the administration of the city of Munich aims to implement a new pricing system for parking lots (PLs). The price of each PL should be calculated based on the number of cars driving in its vicinity. The prices should be updated in a periodic fashion to ensure a fair pricing allocation.

To be able to implement the above mechanism, the city has access to data collected from user cars and PLs belonging to city-run parking stations. For each car, its position and speed are periodically monitored; for each PL its position and availability. Data is stored locally for the course of a full day and submitted for analysis as a full day batch. For illustration, the data sets could look like the ones depicted in Fig. 1.

To get the necessary information of all driving cars near an available PL, the cars data set is filtered to only consider cars with a speed higher than, e.g., 5 km per hour (which indicates that they are not parked). This data set is combined with the PL data sets in order to find cars driving near a PL. This creates a list of PLs and number of cars driving near them. This information is then used to determine the price of a PL.

COMMAND	DESCRIPTION	EXAMPLE
LOAD	Used to load data from the HDFS	A = LOAD 'sample.csv' USING PigStorage(',') AS (name:chararray, age:int, gpa:float);
FOREACH	Run a command for each data row	B = FOREACH A GENERATE name;
GROUP	Groups data into tuples	C = GROUP B BY name;
JOIN	Performs a join on two data sets	N = JOIN A BY name, K BY name;
DISTINCT	Removes duplicate tuples in a relation.	C = DISTINCT B;
CROSS	Creates the cross product of two data sets	F = CROSS C, E;
FILTER	Filters a data set based on an expression	H = FILTER G BY <Boolean expr>

Fig. 2. Popular Pig commands.

B. Background: Pig

Pig Latin [6] is a procedural query language for very large data sets. It offers an alternative to the well-known SQL standard for querying HDFS and is designed to work with the Hadoop infrastructure. Instead of one single relational query, a Pig query consists of a directed acyclic graph of nodes that can be seen as an execution plan. Within this graph, each node describes one step that is needed to execute the query—from loading the data, to the final output. Pig optimizes a query on the fly before sending it to the Hadoop cluster [7] and reaches a performance that is comparable with native Hadoop implementations. It is widely used as an alternative query language and is also included into popular Hadoop distributions like Hortonworks [8].

The Pig language consists of a broad set of commands. The result of each command is assigned to a variable that can be used by other commands in the query. A number of popular Pig commands are depicted in Fig. 2. An important difference to

```
DEFINE Distance
       datafu.pig.geo.HaversineDistInMiles();

-- Create a list of PLs and their position
A = LOAD 'slots.csv' USING PigStorage(',') as
       (PL_ID:int, AVAILABLE:int,
       LONGITUDE:double, LATITUDE:double,
       TIMESTAMP:long);
B = FOREACH A
       GENERATE PL_ID, LONGITUDE, LATITUDE;
C = DISTINCT B;

-- Create a list of all cars that were driving
D = LOAD 'cars.csv' USING PigStorage(',') as
       (CAR_ID:int, SPEED:int, LONGITUDE:double,
       LATITUDE:double, TIMESTAMP:long);
E = FILTER D BY speed > 5.0;

-- Join the data by GPS distance
F = CROSS C, E;
G = FOREACH F GENERATE *, Distance(C::LATITUDE,
       C::LONGITUDE, E::LATITUDE, E::LONGITUDE)
       as DISTANCE;
H = FILTER G BY DISTANCE < 5.0;

-- Count the amount of cars for each PL
I = GROUP H BY PL_ID;
J = FOREACH I {distCars = DISTINCT H.CAR_ID;
       GENERATE $0, COUNT(distCars);};
```

Fig. 3. Possible Pig query for the running example.

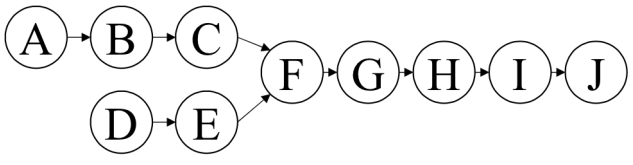


Fig. 4. Abstract graph of the Pig query of the running example.

SQL is that Pig allows special data formats like TUPLE, BAG or MAP. A GROUP command, e.g., puts all elements into a BAG that is associated with the group key. These additional formats can also be used as nested data structures, e.g. a bag within a bag. Additional to the native Pig commands, there is the possibility to use *user defined functions* (UDFs) that implement custom behavior that might require a complex computation. These can be written in Java or Python and included into the Pig query as seen in Fig. 3 (line 1).

For illustration, the running example could be modeled by the Pig query depicted in Fig. 3—the corresponding abstract query graph is depicted in Fig. 4.

III. Tool Description

QryGraph is a tool to simplify the creation, maintenance, evolution, and management of Big Data analytics jobs. An analytics job in QryGraph corresponds to a Pig language query. A Pig query is specified via the use of a graphical editor (the heart of QryGraph), which represents a query by its abstract graph.

The user is able to specify Pig queries via modeling the data flow between nodes corresponding to Pig language commands (see Fig. 2). Each command corresponds to a node in the graph; nodes' input and outputs are connected to create the query graph. The user receives immediate feedback once a type error is introduced in the design process (e.g. when trying to connect a node's output of type A to another node's input of type B). Once a correct query graph is created, the tool automatically compiles it down to valid Pig code and presents it to the user for validation. The generated query can then be issued to a Hadoop cluster.

Apart from the graphical editor, QryGraph provides also an interface for keeping track of all created queries, for issuing in an on-demand or periodic schedule basis, and for notifying the user for the termination of issued queries and presenting the query results.

In the following, we detail on the main design goals of QryGraph, its main usage scenarios, as well as its technical architecture and implementation.

A. Main Design Goals

1) Easy to use

The tool should reduce the cognitive barrier in understanding and creating complex Pig queries. For this, it needs an intuitive and easy to use user interface that helps the novice designer in creating a query (e.g., by offering the possible fields to filter by in a FILTER node). At the same time, it should offer full flexibility to advanced Pig users, who might need to visually inspect or even edit the generated Pig code.

The tool should also provide an ergonomic interface for managing created queries and configuring their scheduling policies.

2) Quick feedback

The tool should offer quick, preferably immediate, feedback when designing a query, so that the designer can resolve type errors early on.

The tool should also provide a "test run" of a query, i.e., execution of the query on a small set of fabricated data instead of production data. This would offer the possibility of semantic checks and validation, answering the question "Does the query actually perform what it is supposed to?"

3) Reuse support

The tool should offer the user the possibility to extract common patterns used across several queries and reuse them in new queries as "composite nodes". One such example could be a composite node that joins GPS position data from two inputs based on a user defined distance function.

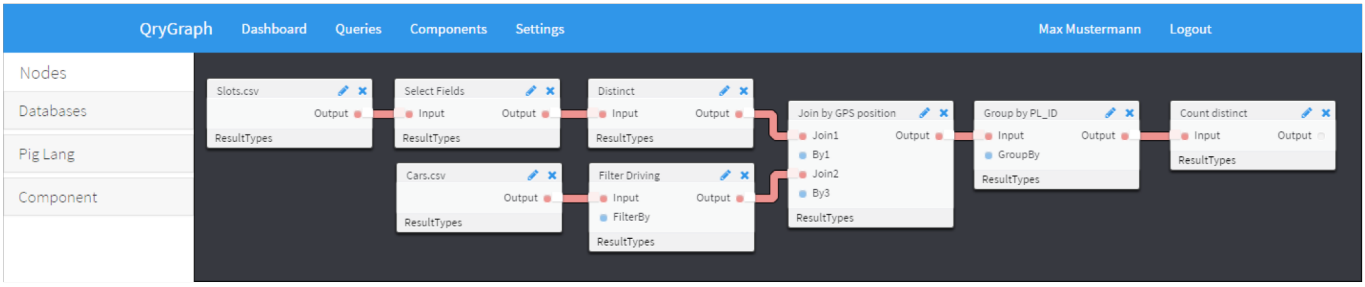


Fig. 5. QryGraph graphical editor.

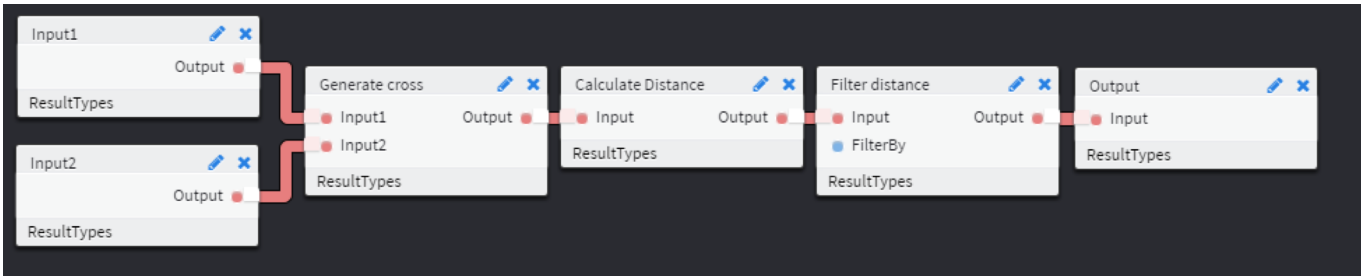


Fig. 6. Subgraph of "Join by GPS position" node.

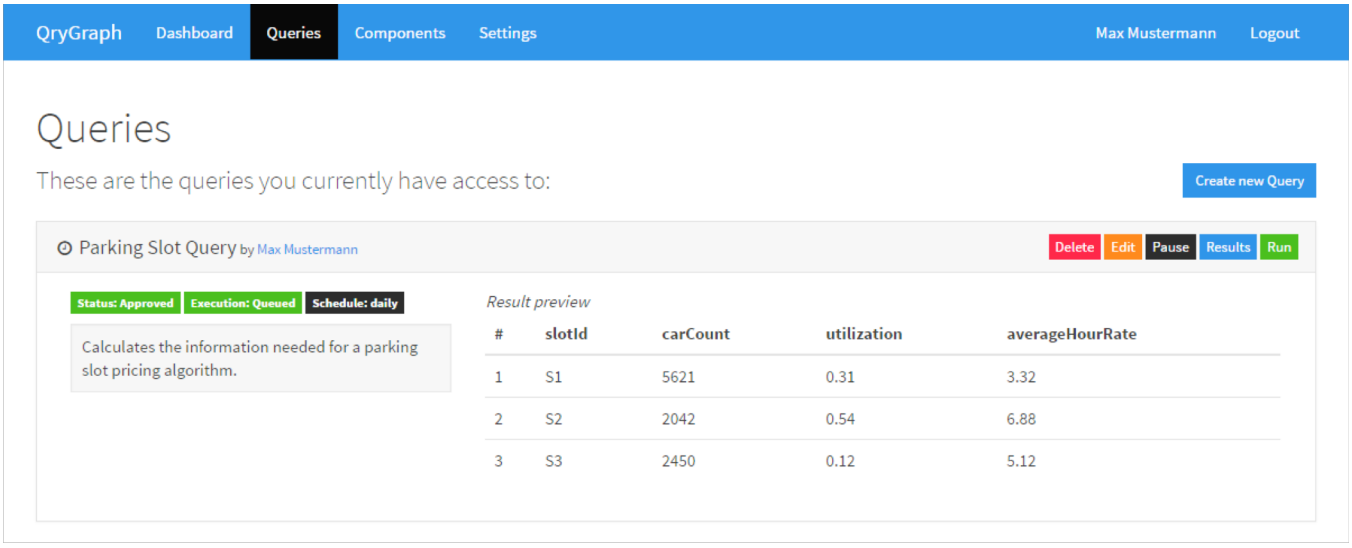


Fig. 7. Query management interface.

4) Easy to setup

The tool should be setup with minimum effort from the user. This will allow prospective users to experiment with the tool and consider contributing by extending its functionalities.

B. Usage

QryGraph offers two main functionalities to its users: query design in a graphical editor and query administration and lifecycle management.

1) Query design

When a user wants to create a new query or edit an existing one, he/she uses the graphical editor. The editor features a command menu on the left side and a big pane to create the query graph on the right side (Fig. 5). Here the user can:

- select the data sources (e.g. files in CSV format) of the query and add them to the graph;
- add native Pig functions (e.g., FILTER, GROUP, JOIN) as nodes to the graph;
- edit the configuration and parameters of the nodes;
- plug nodes together via connecting input ports to output ports;
- automatically get instant feedback on possible type errors after every change;
- inspect the Pig code that is generated on the fly from an error-free graph.

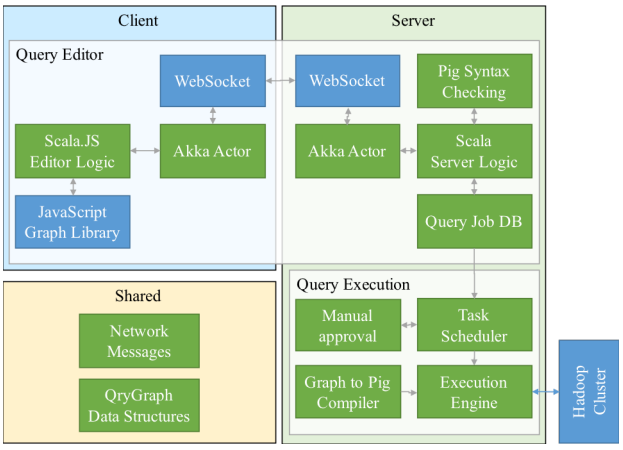


Fig. 8. QryGraph architecture – custom code (green) / libraries (blue).

2) Query administration and management

When a user needs to obtain an overview of the created queries, manage their triggering policies, and view their sample results he/she uses the administrative interface of QryGraph (Fig. 7). Here the tool offers the user to:

- review execution statistics on a dashboard;
- list all queries the user has created;
- pause and suspend query execution;
- change the execution schedule of a query;
- check the approval status of a query (each query has to be approved at the server-side in order to run on production data—see Section III.C);
- view the results of a query.

C. Technical Architecture and Implementation

QryGraph has been implemented as an open-source web application. Its latest version is accessible at https://github.com/Starofall/QryGraph.

The tool follows a client-server architecture (Fig. 8) and is mainly written in Scala. The client component is built with Scala.js[1], a library that compiles Scala into JavaScript. The

[1] https://www.scala-js.org

server is built with the Play Framework[2]; the dynamic server-client communication is done using the actor-oriented Akka Library[3] and Akka.js[4] with underlying web sockets. This connection allows to update the client on compilation results without using any long polling technique. Using the abstraction layer provided by Akka actors, the client and server communicate via typed messages sent serialized through the socket. The message classes and data structures are shared by the server and client Scala code. This simplifies development and debugging since a node instance has the same behavior on both the client and the server and no manual data parsing on the server or the client is required.

In summary, the internal workings of the graphical editor are the following: Once a user edits an element of a graph, the graph object is serialized and sent over the network to the server. Then the server performs syntax checking. If the graph is syntactically correct, Pig code is generated out of it and sent back to the user for visual inspection. Once the user wants to issue the query to the Hadoop cluster, the execution request is send to the server where it is added to the queue of queries that need manual approval by the cluster administrator (an optional step in the process). Then, the query is issued to the Hadoop cluster. Once the results are computed, they are sent back to the user via the same actor-based communication.

IV. Discussion and Work in Progress

QryGraph is a project under active development. In the initial stages documented in this paper, we were mainly focusing on implementing the programming abstractions and overall infrastructure that would allow to speed up feature development. For instance, we now have a very robust actor-based client-server communication that allows for seamless development.

In the following, we reflect on the main architectural decisions and on the features under implementation.

A. Web-Based

The QryGraph client has been implemented as a web-based application. One of the primary drivers for doing so was the ease of use, as users are typically familiar with browser environments. At the same time, this allows the developers to use many off-the-shelf libraries that are available for JavaScript and CSS (e.g. Twitter Bootstrap).

The web-based environment is also making the tool setup easy: it is only required to deploy a JVM-based application on a server. Then any user with a modern browser is able to start using the tool and create queries.

B. Seamless Client-Server Development

In order to leverage on the domain-specific modeling capabilities and the type system of Scala, we opted for using Scala.js to generate most of the JavaScript code, in particular the parts related to the representation of queries on the client side. Using a well-known Scala library for actors, Akka, and its Scala.js counterpart, Akka.js, allows developers to work with the same actor-based abstractions on both client and server side. The communication is conveniently abstracted into typed messages sent between Scala actors over a web sockets connection.

C. Enhanced Testing and Debugging – In Progress

At its current state, the graphical editor gives instant feedback to the user, as changes are being made on the graph, regarding syntax errors. This is performed at the server side by type checking (e.g. checking that connected inputs and outputs are of the same type, GROUP operators operate on attributes given as inputs, etc.). This offers a mechanism to spot errors early on and refactor the query.

An additional mechanism (under development) is to allow sample runs of an error-free query for debugging reasons. Running a query on the production data can be a lengthy process taking minutes or hours to complete. However, for debugging purposes, a much quicker response has to be provided to the user. To this end, a test data set has to be provided against which test runs can be issued. Providing such a test data set is far from easy and typically has to be tailored to the particular user program at hand [6]. This data set has to be considerably smaller than the production data set, yet still realistic; it can be used for two types of tests:

- *Fit-for-purpose validation*, i.e. checking whether the query has the intended effect;
- *Performance* testing: When performing multiple queries to the same test data set or test data sets of approx. the same size, the execution times of past runs can be used as an indicator of a low-performing query. Such analysis is based on the fact that the relative difference in the execution times of queries is of importance, and not the actual values.

D. Component System – In Progress

One of the advantages of the graphical editor is that it makes the data-flow architecture of a query explicit, which improves program comprehension and maintainability. To enable reuse of query fragments (subgraphs) that are common across different queries, we are working on providing a mechanism that allows the creation of components with well-defined inputs and outputs out of query subgraphs. These components would then be used as regular nodes in the graphical editor, similar to black-box component composition.

An example of such a component could be a subgraph that joins GPS position data from two inputs based on a user defined distance function. The creation of these component will be done via the main graphical editor, as illustrated in Fig. 6.

V. Related Work

Due to the recent advancements in Big Data infrastructures and tools and in the Hadoop ecosystem in particular, many different tools and products have been proposed. We focus here on tools that offer graphical interfaces for viewing or specifying Big Data analytics jobs, thus are directly comparable to QryGraph.

The open-source project PigPen [6], [9] is an extension to the Eclipse platform that allows the user to specify Pig queries in a

[2] https://www.playframework.com
[3] http://akka.io
[4] https://github.com/unicredit/akka.js

textual editor and then inspect the corresponding query graph, which is updated on the fly. However, editing the graph is not supported. PigPen also offers error checking, and running a query in a "sandbox" data set for debugging. We intend to reuse their approach in the creation of our test data set infrastructure.

A mature open-source solution, also based in Eclipse, is Talend Open Studio for Big Data [10]. It allows the specification of Big Data analytics jobs in a graphical interface. Compared with this tool, QryGraph is more lightweight and requires less upfront effort for setting up and getting started with the tool.

When looking at closed-source solutions, Tableau Desktop [11] is a tool that offers an easy-to-use user interface for specifying and running Big Data analytics. Instead of supporting the generation of a specific query, this tool focusses on data visualization. Pentaho [12] is offering several closed source products that also help the user define a data flow using a graphical interface. They support multiple database configurations.

VI. Conclusions

In this artifact paper, we presented our ongoing implementation and long-term plan for a new tool for simplifying the creation and management of Big Data analytics jobs. QryGraph focuses on the popular scripting language Pig and offers visual representation and editing of Pig queries. It allows for rapid prototyping by on-the-fly compilation and syntax checking, and promotes reuse by allowing for specifying user-defined query subgraphs.

One of the features that we would like to include in QryGraph in the future is the possibility for importing existing Pig queries into the graphical editor. This will allow for creating a library of example queries for educational purposes and of subqueries that can be reused.

Finally, we would like to support cooperative query editing. The existing implementation based on web sockets already provides the base upon which several users can work locally and communicate with the server with independent query updates.

Acknowledgment

This work is part of the TUM Living Lab Connected Mobility project and has been funded by the Bayerisches Staatsministerium für Wirtschaft und Medien, Energie und Technologie.

References

[1] J. Needham, *Disruptive Possibilities: How Big Data Changes Everything*. O'Reilly Media, 2013.

[2] E. Dumbill, *Planning for Big Data*. O'Reilly Media, 2012.

[3] S. Srinivasa and V. Bhatnagar, Eds., *Big Data Analytics*, vol. 7678. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[4] K. Shvachko, H. Kuang, S. Radia, and R. Chansler, "The Hadoop Distributed File System," in *Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, Washington, DC, USA, 2010, pp. 1–10.

[5] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *Commun. ACM - 50th Anniv. Issue*, vol. 51, no. 1, pp. 107–113, Jan. 2008.

[6] C. Olston, B. Reed, U. Srivastava, R. Kumar, and A. Tomkins, "Pig Latin: A Not-so-foreign Language for Data Processing," in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, New York, NY, USA, 2008, pp. 1099–1110.

[7] A. F. Gates, O. Natkovich, S. Chopra, P. Kamath, S. M. Narayanamurthy, C. Olston, B. Reed, S. Srinivasan, and U. Srivastava, "Building a High-level Dataflow System on Top of Map-Reduce: The Pig Experience," *Proc VLDB Endow*, vol. 2, no. 2, pp. 1414–1425, Aug. 2009.

[8] "Hortonworks," 01-May-2016. [Online]. Available: http://hortonworks.com/.

[9] "PigPen Wiki," 01-May-2016. [Online]. Available: https://wiki.apache.org/pig/PigPen.

[10] "Talend Open Studio for Big Data," 01-May-2016. [Online]. Available: https://de.talend.com/download/talend-open-studio.

[11] "Tableau Desktop," 01-May-2016. [Online]. Available: http://www.tableau.com/de-de/products/desktop.

[12] "Pentaho," 01-May-2016. [Online]. Available: http://www.pentaho.com/product/data-integration.

Rethinking High Performance Computing System Architecture for Scientific Big Data Applications

Yong Chen*, Chao Chen*, Yanlong Yin†, Xian-He Sun†, Rajeev Thakur‡, William D Gropp§

*Department of Computer Science, Texas Tech University, Email: yong.chen@ttu.edu, chao.chen@ttu.edu
†Department of Computer Science, Illinois Institute of Technology, Email: yyin2@ttu.edu, sun@ttu.edu
‡Mathematics and Computer Science Division, Argonne National Laboratory, Email: thakur@mcs.anl.gov
§Department of Computer Science, University of Illinois Urbana-Champaign, Email: wgropp@illinois.edu

***Abstract*—The increasingly important data-intensive scientific discovery presents a critical question to the high performance computing (HPC) community - how to efficiently support these growing scientific big data applications with HPC systems that are traditionally designed for big compute applications? The conventional HPC systems are computing-centric and designed for computation-intensive applications. Scientific big data applications have growlingly different characteristics compared to big compute applications. These scientific applications, however, will still largely rely on HPC systems to be solved. In this research, we try to answer this question with a rethinking of HPC system architecture. We study and analyze the potential of a new decoupled HPC system architecture for data-intensive scientific applications. The fundamental idea is to decouple conventional compute nodes and dynamically provision as data processing nodes that focus on data processing capability. We present studies and analyses for such decoupled HPC system architecture. The current results have shown its promising potential. Its data-centric architecture can have an impact in designing and developing future HPC systems for growingly important data-intensive scientific discovery and innovation.**

I. Introduction

Many scientific simulations in critical areas, such as climate sciences, astrophysics, computational chemistry, computational biology, and high-energy physics, are becoming increasingly data intensive [1, 2]. These applications manipulate a large amount of data relative to the amount of computation they perform, and often transfer large amounts of data to and from storage systems. Some application teams have already begun to process terabytes or tens of terabytes of data in a single simulation run. For example, 12 out of 25 INCITE applications run on the Department of Energy leadership computing system at Argonne National Laboratory several years ago have already processed datasets in the terabyte range [3, 4].

Meanwhile, the data collected from instruments for scientific discoveries and innovations are increasing rapidly too. For example, the Global Cloud Resolving Model (GCRM) project, part of Department of Energy's Scientific Discovery through Advanced Computing (SciDAC) program [5], is built on a geodesic grid that consists of more than 100 million hexagonal columns with 128 levels per column. These 128 levels will cover a layer of 50 kilometers of atmosphere upwards from the surface of the earth. For each of these grid cells, scientists need to store and predict data like the wind velocity, temperature, pressure, etc. Most of these global atmospheric models currently process data in a 100-kilometer scale (the distance on the ground); however, scientists desire higher resolution and finer granularity, which can lead to even larger sizes of datasets. In addition, the proliferation of sensing technologies and the increased usage of remote sensors are also generating huge amount of data than ever before.

Both experimental data and simulation data are rapidly increasing, and these scientific discoveries and innovations have exhibited a critical data-intensive computing need, creating the "big data" computing era in recent years [6–8]. High Performance Computing (HPC) is a strategic tool during the process of scientific discoveries and innovations. These increasingly data-intensive scientific problems will still rely on HPC systems to compute, analyze, and answer the problems, which is an essential process of understanding the phenomenon behind the data. The conventional HPC systems, however, are *computing-centric* and designed for *computation-intensive applications*. They are not ready and can have inherent limitations when used for solving increasingly data-intensive problems. How to design and develop HPC system architectures for efficient processing ever-growing scientific data has become a key challenge in the big data computing era.

In this research, we revisit the HPC system architecture and study the impact of a new *decoupled high performance computing system architecture* for data-intensive sciences. Such a study can shed light on designing and developing next-generation HPC systems for growingly critical data-intensive computing. A decoupled system architecture has the notion of the separation of compute nodes and data processing nodes. The novelty of the decoupled HPC system architecture is that, instead of dedicating the dominant investment to compute nodes as in the conventional HPC system architecture, the new investment is decoupled into compute nodes and data nodes. These data nodes are designed to handle data-intensive operations with a mission of minimizing data movement. Compute nodes, as in the conventional architecture, handle computation-intensive operations. Scientific big data applications can be mapped to such a decoupled HPC system architecture and are executed in a decoupled but fundamentally more efficient manner with the collective support from data nodes and compute nodes. Ideally these data nodes and compute nodes can be dynamically configured and provisioned depending on application-specific characteristics, e.g. the intensity of data accesses. This research focuses on modeling and analysis of a decoupled HPC system architecture. A fundamental question we try to answer in this research is that how HPC system architecture should be designed and developed to best support data-intensive scientific computing. Our idea of a decoupled system architecture is a new thinking of HPC system architecture design and development when

2324-9013/16 $31.00 © 2016 IEEE
DOI 10.1109/TrustCom.2016.246

data access is as important as computation. A decoupled HPC system architecture can be such a possible solution. The current results have shown that it is promising and has a potential.

The rest of this paper is organized as follows. Section II presents the idea and framework of a decoupled HPC system. Section III introduces the modeling and analysis of the decoupled HPC system architecture. Section IV reviews existing studies in related areas and compares with our work. Section V concludes this study and discusses future work.

II. Decoupled High Performance Computing System Architecture: Motivation and Overview

In scientific applications, data is commonly represented with a multi-dimensional array-based data model. For instance, the widely used Community Earth System Model (CESM) software package consists of four separate modules simultaneously simulating the earths atmosphere, ocean, land surface and sea-ice, and each module uses the multi-dimensional arrays data model [9]. A common example is a 3-dimensional temperature data with longitude, latitude, and time dimensions. It is often needed to compute the moving average, median, lowest and highest temperature with specified conditions such as areas and periods of time. Such computed results will be further correlated with the computed results from other parameters, such as the humidity and wind velocity, to predict weather conditions.

The current way of conducting such processing is to read the required data (e.g., a sub-array of interested area) from storage servers to compute nodes, perform computations on desired data with specified conditions, such as those data shown in shaded area, and then write the output back to storage. For CESM, an experimental test shows that the data access and movement time for the calculation of the moving average, median, lowest and highest degrees can occupy 88.2%, 95.4%, 96.6%, and 96.6% of the total execution time on a cluster, where 128GB of data are retrieved to 272 nodes for processing.

CESM clearly has data retrieval and processing phases and computing and simulation phases, as many scientific big data applications do. The basic idea of the new decoupled HPC system architecture is to change the conventional architecture to handle these two phases differently on different nodes. Such an architecture decouples nodes into compute nodes and data processing nodes. These nodes are mapped with computation-intensive operations and data-intensive operations respectively. Computation-intensive operations are executed on massive compute nodes. Data-intensive operations are executed on dedicated data processing nodes. In other words, the decoupled architecture reshapes the current pattern of retrieve - compute - store cycles into retrieve (generate) - reduce - compute - reduce - store cycles as shown in Figure 1, where the reduce phases are designed to conduct offloaded data-intensive operations and reduce data size before moving data across the network. These retrieval, reduce, compute, and store phases can be pipelined to overlap the I/O, communication, and computation times. From one point of view, the decoupled architecture is an enhanced framework of MapReduce [10], where the reduce is not conducted by one node with its local storage, but a set of (data) nodes and the global storage, so that parallel computing features can be maintained. From another point of view, the data nodes are the data-access accelerators, to speed up data accesses and reduce data size before sending data across the network.

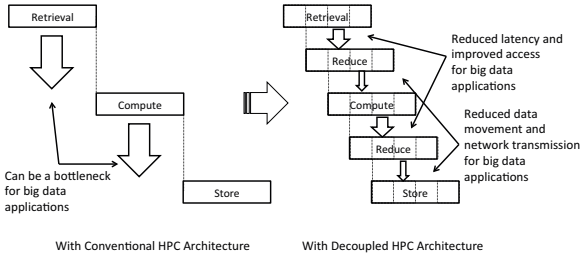


Fig. 1. Comparison of HPC Architectures

The decoupled HPC system architecture is shown in Figure 2. This architecture decouples the nodes of conventional investment into data nodes and compute nodes. Data nodes are further decoupled into compute-side data nodes and storage-side data notes. Compute-side data nodes are compute nodes that are dedicated for data processing. Storage-side data nodes are specially designed nodes that are connected to file servers with fast network. Compute-side data nodes reduce the size of computing generated data before sending it to storage nodes. Storage-side data nodes reduce the size of data retrieved from storage before sending it to compute-side data nodes. Writes will go through compute-side data nodes, whereas reads will go through the storage-side data nodes. Data nodes can provide simple data forwarding without any data size reduction, but the idea behind data nodes is to let data nodes conduct the offloaded data-intensive operations and optimizations to reduce the data size and data movement.

In this research, we focus on studying the implications of such a decoupled HPC system with an assumption that operations from applications can be decoupled into computation-intensive and data-intensive operations respectively. Our study focuses on how to design and configure the decoupled HPC system architecture considering scientific big data applications features, such as the intensity of data accesses and characteristics of computing and data accesses. Performance tools can provide information and guidance to understand application computing and data-access characteristics and intensity of data accesses. For instance, we have developed an IOSIG performance tool to find the I/O access signature (patterns) of an application [11, 12]. We have extended IOSIG to IOSIG+ to identify the data-intensive phases and computation-intensive phases. I/O dependency analysis can also be used to separate the phases [13], and can be used to find the hot data which lead to operations that should be conducted on the data nodes.

The decoupled HPC system architecture changes the current architecture by balancing the computation and data-access capabilities for data-intensive applications. This new architecture separates computation-intensive operations and data-intensive operations and handles them concurrently and in a coordinated manner, but on different hardware and software environments for best performance. The architecture configuration is flexible. At one extreme, it could have no data nodes. In that case, it is the traditional HPC architecture. At another extreme, the compute nodes could be simple SIMD processing elements. In that case, the compute nodes are more

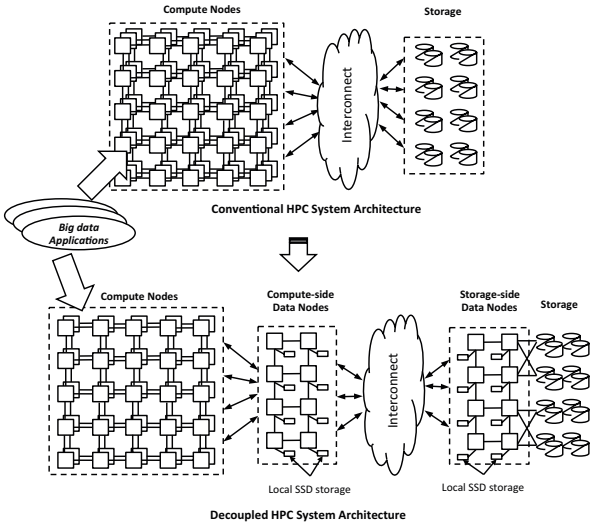


Fig. 2. Decoupled High Performance Computing System Architecture

like computing accelerators. In this research, while we will study different configurations, the focus is on understanding such decoupled system architecture and impact on scientific big data applications. We will discuss details in the following section from modeling and analyses.

III. Decoupled High Performance Computing System: Modeling and Analysis

In this section, we present detailed modeling and analyses of the decoupled HPC system. We first introduce the performance model used to analyze the potential of system architecture designs, and then present detailed studies from both architecture configurations and also application features.

A. Modeling and Comparison of HPC Systems

There is no argument that many performance models for parallel computing and HPC systems exist, such as the well-known Amdahl's model. A common limitation of these models, however, is that they primarily focus on computation part of applications to direct building HPC systems for computation-intensive applications. Our analyses are based on a simple but effective model that captures both computation and data access, which is especially needed for analyzing HPC system architectures for data-intensive applications given the growing importance.

Different from previous models, our model takes data workload into consideration to analyze the system performance. This model assumes that the execution of an application logically follows the model illustrated in the Figure 3, which is a simplified abstraction and generally holds for many applications without optimizations like pipelining, asynchronous I/O, etc. This performance model is based on an assumption that applications conduct interleaved and periodic computations and data accesses. In each phase, the total workload W contains two parts: computation workload part (W_C) and data workload part (W_D). For simplicity, we assume that W_C and W_D are the same for each phase. The model and analysis can be applicable to a general case that W_C and W_D are different for each phase, which means that there exist W_Ci and W_Di for each phase. Thus, the entire workload of an application can be derived and expressed as:

$$W = (W_C + W_D) \cdot m \tag{1}$$

where m is the number of phases.

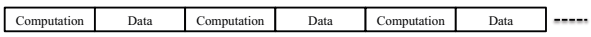


Fig. 3. Execution Model of An Application on High Performance Computing Systems

To derive the detailed analysis of the new decoupled HPC system architecture, the model also introduces several notations to characterize dominant applications and systems parameters, as shown in Table I.

TABLE I
Parameter Notations

n	the number of compute nodes
r	the ratio of data nodes compared to compute nodes
b	the network bandwidth of each compute node in the conventional and decoupled architectures
b_h	the network bandwidth of each data node in the decoupled system architecture
λ	the ratio between the network bandwidth of compute nodes and data nodes, i.e. b_h/b
$f_{op}(x)$	the result workload of an offloaded data-intensive operation op with input x raw data workload, or the computation workload (termed as *seed workload*) for generating x size workload
η	the ratio of data-intensive operation's computation workload compared to the whole computation workload of an application
γ	the ratio of the result workload or seed workload of an offloaded data-intensive operation compared to the raw data workload
α	the ratio between the computation workload and data workload

In conventional HPC system architecture, the total workload is divided and executed on n compute nodes. Applications process data on compute nodes and conduct I/O operations directly with storage nodes. Thus, the execution time (T) and the performance (P) of a conventional HPC system can be modeled as:

$$T = (\frac{W_C}{n} + \frac{W_D}{n \cdot b}) \cdot m \tag{2}$$

$$P = \frac{1}{T} \tag{3}$$

With the decoupled HPC system architecture, data nodes (including compute-side data nodes and storage-side data nodes) are specially designed for data-intensive operations. Data nodes are connected to compute nodes (for compute-side data nodes) or storage nodes (for storage-side data nodes) through a high-speed network. Data-intensive operations can be decoupled and offloaded to data nodes according to their features. In a read-intensive situation, where the application retrieves a large volume of data and then performs computations (such as SUM and k-means) to obtain a small-size result for further processing, this operation can be offloaded to storage-side data nodes and only the result is returned to compute nodes. The decoupled system architecture reduces considerable data movement and improves the usage of the precious network bandwidth. In a write-intensive situation, the computations that generate a large volume of data (we term as "seed operations") can be offloaded to storage-side data nodes.

These computations are performed on storage-side nodes to generate data in place instead of generating data on compute nodes and moving through the network. The execution time and the performance of the decoupled system architecture can thus be derived and defined as

$$T' = (\frac{W_C - W_{op}}{n\cdot(1-r)} + \frac{f_{op}(W_D)}{n\cdot(1-r)\cdot b} + \frac{W_D}{n\cdot r\cdot b_h} + \frac{W_{op}}{n\cdot r}) \cdot m \quad (4)$$

$$P' = \frac{1}{T'} \quad (5)$$

where W_{op} is the related computing workload of the data-intensive operation. It is directly related to data workload and can not be ignored. In the model, it can be expressed as $W_{op} = \eta \cdot W_C$. In practice, parameters, η, γ, λ, meet the following conditions:

$$0 < \eta < 1 \quad (6)$$

$$0 < \gamma < 1 \quad (7)$$

$$\lambda > 1 \quad (8)$$

To compare the performance of the decoupled HPC system architecture and conventional architecture, we can calculate the performance difference, defined as:

$$\Delta = P' - P = \frac{T - T'}{T \cdot T'} \quad (9)$$

If $\Delta > 0$, it means that the decoupled system architecture has better performance than the conventional architecture. Otherwise, the conventional architecture is better. Due to $T \cdot T' > 0$, the sign of the Δ will depend on $\rho = T - T'$. Based on the above assumptions and analyses, the equation for calculating ρ is derived as:

$$\rho = \frac{m}{n\cdot(1-r)\cdot r} \cdot [(2\cdot r\cdot\eta - \eta - r^2)\cdot W_C + \frac{\lambda\cdot r\cdot(1-r) - \gamma\cdot r\cdot\lambda - 1 + r}{\lambda\cdot b} \cdot W_D] \quad (10)$$

Another parameter, α, is introduced to represent the relationship between W_C and W_D and is defined as:

$$W_C = \alpha \cdot W_D \quad (11)$$

Equation (11) can be used to quantify the data-access intensiveness of an application. If $\alpha > 1$, which means the computation workload is larger than the data workload, the application can be considered as computation-intensive. Otherwise, if $0 < \alpha < 1$, the application can be considered data-intensive. With the parameter α, ρ can be derived as:

$$\rho = \frac{m\cdot W_D}{n\cdot(1-r)\cdot r} \cdot [(2\cdot r\cdot\eta - \eta - r^2)\cdot\alpha + \frac{\lambda\cdot r\cdot(1-r) - \gamma\cdot r\cdot\lambda - 1 + r}{\lambda\cdot b}] \quad (12)$$

In equation (12), all parameters, including both system-related and application-related parameters, are involved. This observation confirms the expectation that different applications need different system configurations to achieve the best performance. In general, given a system and an application, m, n, W_D are constant values, and they can not affect the system configurations. The rest of parameters will impact the system configurations. Thus, ρ' is introduced as follows and the rest analysis focuses on understanding the impact of these parameters of both system and applications' characteristics on the performance:

$$\rho' = \frac{1}{(1-r)\cdot r} \cdot [(2\cdot r\cdot\eta - \eta - r^2)\cdot\alpha + \frac{\lambda\cdot r\cdot(1-r) - \gamma\cdot r\cdot\lambda - 1 + r}{\lambda}] \quad (13)$$

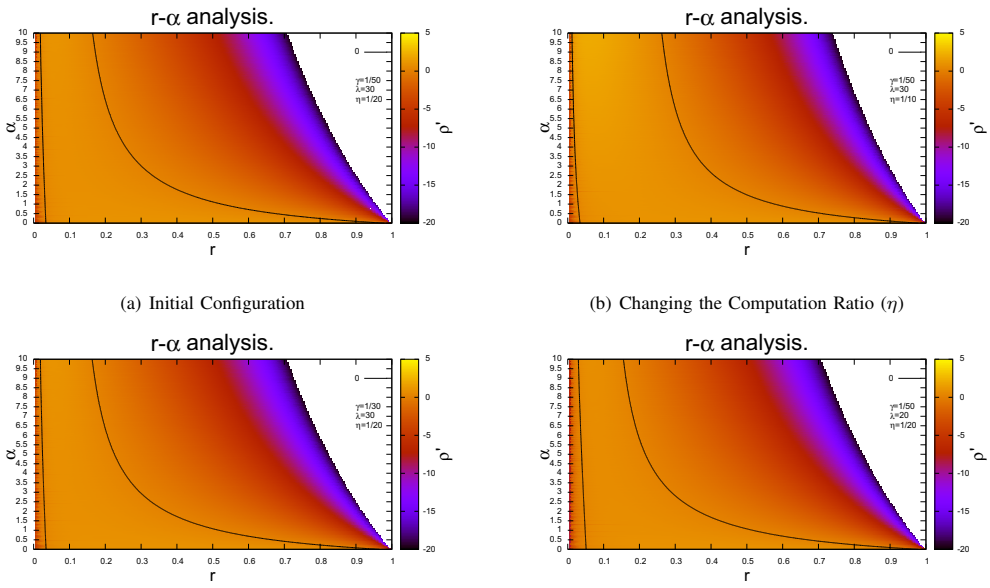
B. Analysis of the Decoupled HPC System Architecture

In this subsection, we focus on using the performance model discussed in the previous subsection to evaluate whether the new decoupled system architecture is more effective than the conventional architecture for scientific big data applications. This evaluation specifically focuses on two parameters, r and α, because they represent the system architecture and applications characteristics respectively. First, r denotes the ratio of the data nodes in the decoupled system architecture. Different values of r represent different system configurations. For example, if $r = 0$, there will be no data nodes in the system, and the architecture will be completely the same with the conventional architecture. Second, α in the model is used to measure the intensity of data accesses in an application. If $0 < \alpha < 1$, it implies more data workload than computation workload; thus, the application is more data-intensive. Otherwise, if $\alpha > 1$, it implies that the application is more computation-intensive. Therefore, α is the parameter that characterizes applications features.

The goal of this evaluation is to compare the decoupled HPC system architecture and the conventional architecture. Variable ρ' is used to evaluate which one is better. If $\rho' < 0$ holds under any configuration for r and α, it means that the conventional architecture outperforms the decoupled architecture for both data-intensive and computation-intensive applications. Otherwise, it confirms that the decoupled architecture achieves better performance than the conventional architecture under evaluated configurations of systems and applications.

Figure 4 reports the impact of r and α on performance. It plots four $r - \alpha$ graphs given various settings, where r varies from 0 to 1, and α varies from 0 to 10. To better compare the decoupled system architecture and the conventional architecture, these four graphs were plotted with different configurations for λ, η, and γ, as indicated on each graph. Different colors are used to represent different values of ρ'. To assist comparison, we specifically draw a contour line 0, which represents $\rho' = 0$, to help identify these areas that represent $\rho' > 0$ or $\rho' < 0$. The contour line 0 splits each graph into three areas: the left-most and right-most areas that represent $\rho' < 0$, and middle area that represents $\rho' > 0$. The left-most area is caused by the model derivation, because when $r \to 0$, $\frac{W_D}{n\cdot r\cdot b_h}$ in T' will be sufficiently close to ∞, which is impossible to happen in practice.

Comparing Figure 4-a and Figure 4-b, we can find that when the value of η increases, the space of middle area increases too, which means the decoupled system architecture is beneficial for more cases. This is because when η increases, the computation workload conducted on compute nodes will be decreased. Figure 4-c plots the results when increasing value γ. Parameter γ represents the offloading efficiency of a data-intensive operation in reducing the data movement. The lower the the value of γ is, the more data reduction and the less time in transferring data. Compared to Figure 4-a, it shows that γ has little impact on the system configuration. It is not necessary to reconfigure the ratio of data nodes when



(a) Initial Configuration

(b) Changing the Computation Ratio (η)

(c) Changing the Ratio of Result Workload (γ)

(d) Changing the Ratio of Network Bandwidth (λ)

Fig. 4. Analysis of the Decoupled High Performance Computing System Architecture[1]

γ is changed. Similarly, Figure 4-d plots the results when λ is changed, which has minor impact on system configuration as well.

In summary, despite the model derivation, the results shown in Figure 4 illustrate that the decoupled HPC system architecture is better than the conventional architecture for data-intensive applications in most cases (where $\alpha < 1$). Besides, the results also show that when $\alpha > 1$, the decoupled architecture can still be configured with suitable ratio of data nodes to achieve better performance than the conventional architecture. In addition, these graphs imply that the ratio of data nodes should be configured according to applications' characteristics. When applications tend to be more computation-intensive, the number of data nodes should be reduced accordingly. As observable from the graphs, when the α increases from 0 to 1, the value of r is decreased from 1 to around 0.55. When the α increases from 1 to 10, the value of r is decreased slowly, from 0.55 to 0.23. A dynamically configurable architecture would be an ideal solution to best support applications with considering applications' characteristics.

C. Analysis from Systems' Perspective

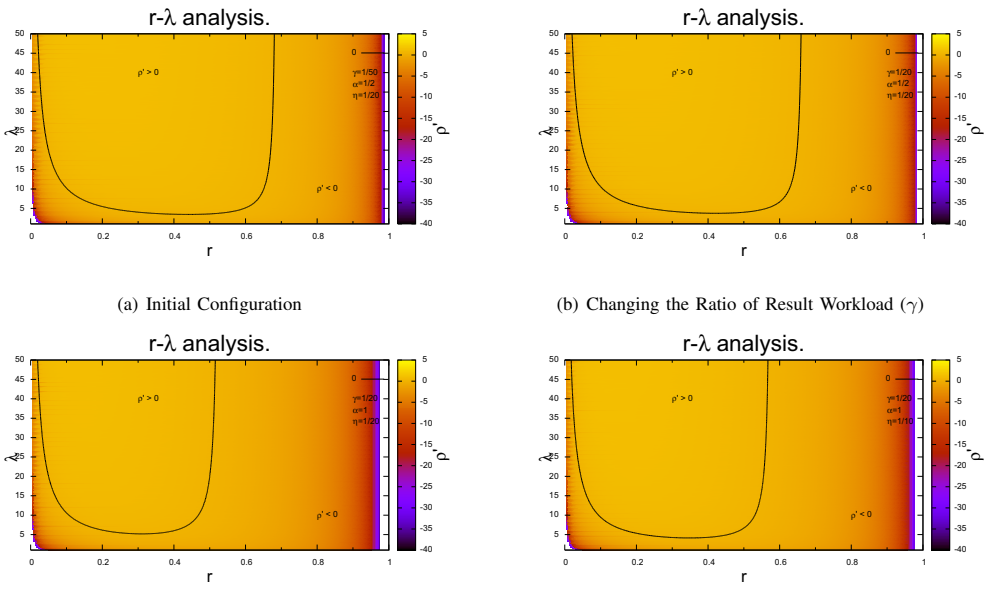
As analyzed in subsection III-B, a dynamically configurable system architecture is a preferred solution. In practice, such an ideal solution is hard to be achieved. One critical challenge is that, modern high performance computing systems are designed to run many scientific applications simultaneously. A dynamic configuration for a specific application would be challenging for other applications. Therefore, how do we design and develop a fixed system configuration for various data-intensive applications? In this subsection, we try to answer this question. We will analyze and show how to configure a high performance computing system with the decoupled architecture for scientific big data applications without prior knowledge of applications.

To answer this question, we need to analyze the system-related parameters. In the performance model, r and λ are two system-related parameters. Parameter r represents the ratio of data nodes configuration. It illustrates how to deploy data nodes and compute nodes. Parameter λ represents the network configuration. It shows the network requirement of a decoupled system architecture. In practice, λ is completely determined by the interconnection deployed and physical configuration. If data nodes are deployed physically closer to storage node, the λ is expected to be higher.

Since we do not have the prior knowledge and need to support multiple applications, we use different applications to find an overlap area to determine the optimal values for r and λ. Figure 5 plots four $r - \lambda$ graphs with changing values of application related parameters, including γ, α, and η. We varied the values of these parameters one by one, and observe how these parameters can affect the system configuration. Various values of γ, α, and η represent different applications. Since we focus on data-intensive applications, we make α vary between 0 and 1.

In general, these figures show that, the decoupled HPC system architecture can achieve better performance for scientific big data applications when the values of λ and r range in an enclosed area. As can be observed from the graphs, when γ increased from $\frac{1}{50}$ to $\frac{1}{20}$, the configuration for r and λ only changes by a small amount, which is consistent with the analysis in subsection III-B. Besides, when the value of α increases from $\frac{1}{2}$ to 1, the preferred range for r shrinks from $(0.1, 0.63)$ to $(0.1, 0.5)$. This observation is reasonable, because when α increases, the application tends to be more computation-intensive and it needs more compute nodes to

[1]We plotted in color for the best result. Please read from the electronic file or color printouts. The rest of figures were plotted in a similar setting.



(a) Initial Configuration

(b) Changing the Ratio of Result Workload (γ)

(c) Changing the Ratio of Computation Workload (α)

(d) Changing the Ratio of Network Bandwidth (λ)

Fig. 5. Analysis from Systems' Perspective

conduct computations. In addition, if we keep increasing the value for η, the preferred range for r will expand slightly, because the computation workload conducted on compute nodes would be decreased as η increases.

In summary, we can conclude that, when designing and deploying a high performance computing system for data-intensive applications, there are two rules we can use from the study of this research:

1) Ensure $\lambda \cdot r > 1$
2) Choose a ratio of data nodes in the range of $(0.2, 0.4)$

D. Analysis from Applications' Perspective

In the performance model, three parameters are used to represent applications' characteristics: α, γ, and η. Parameter α is used to quantify the data-access intensity of an application and identify whether an application is data-intensive ($\alpha < 1$) or computation-intensive ($\alpha > 1$). Parameter γ is used to identify the offloading effectiveness of the data-intensive operation in reducing the data size. Parameter η represents the computation workload of data-intensive applications.

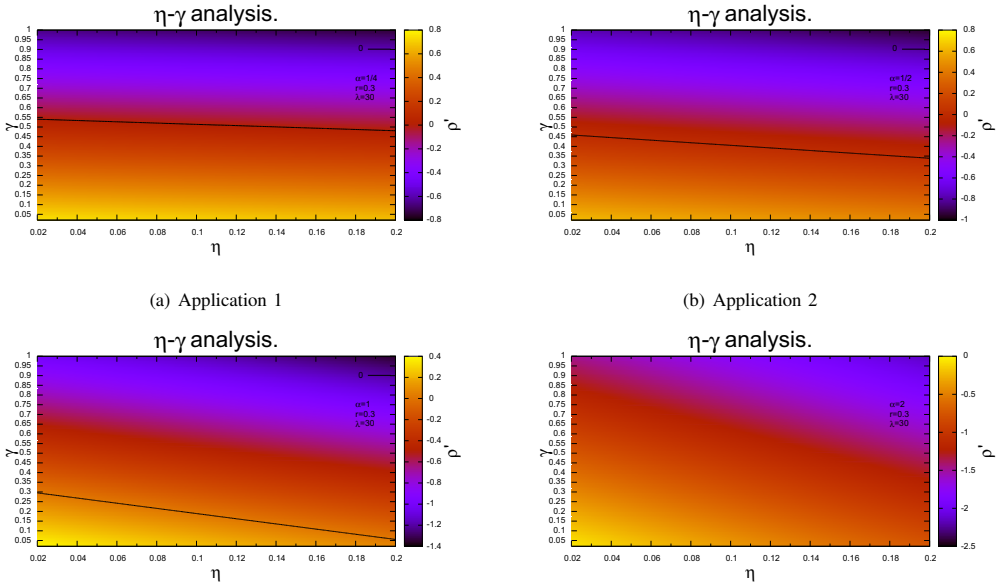
In subsection III-B, we have made several conclusions about the relationship between r and α. We conclude that the decoupled HPC system architecture is a better solution for data-intensive applications. In addition to α, two other parameters, γ and η, also represent applications' characteristics. Even though we find that both parameters have minor impact on system configurations through the earlier analysis, we present a more detailed analysis in this subsection. We attempt to answer the question: can the new decoupled HPC system architecture be deployed for all scientific big data applications?

To answer this question, we evaluated four different applications with different intensity of data accesses (indicated by the value of α). A fixed system configuration is used for this evaluation, and the values for system-related parameters were chosen based on the analysis in the prior subsection. Figure 6 plots four $\eta - \gamma$ graphs to represent four different applications respectively. These figures show that, compared to η, parameter γ has a greater impact on the system configuration. Although changing slightly according to different data-access intensity, it generally requires $\gamma < 0.3$ for most data-intensive applications. This observation means that applications with data-intensive operations that can reduce the data size to lower than $\frac{3}{10}$ of the raw data size can especially benefit from the decoupled HPC system architecture. For computation-intensive applications (Application 4 in the figure), the configuration of the decoupled system architecture in this evaluation did not provide better performance than the conventional architecture. However, according to the analysis in subsection III-B, we can configure a lower value of r for computation-intensive applications to improve the performance with the decoupled system architecture, as shown in Figure 7. It restricts the values of η and γ though, especially for η. In fact, this observation is straightforward because when an application becomes more computation-intensive, the value of η will reduce automatically.

In summary, scientific big data applications that have low computation ratio ($\eta < 0.2$) and can reduce the data size effectively ($\gamma < 0.3$) can especially benefit from the decoupled HPC system architecture. Meanwhile, computation-intensive applications can also benefit from the decoupled system architecture with choosing a proper configuration ratio, which confirms the conclusion in subsection III-B.

IV. Related Work

Extensive studies have focused on improving the performance of HPC systems for data-intensive applications at various levels. At the hardware level, the emerging nonvolatile storage-class memory (SCM) devices such as flash-memory based solid-state drives (SSDs) and phase-change memory



(a) Application 1

(b) Application 2

(c) Application 3

(d) Application 4

Fig. 6. Analysis from Applications' Perspective

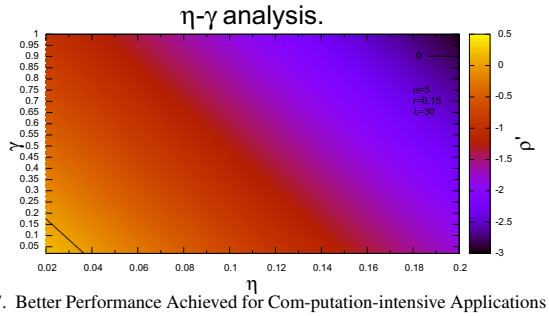


Fig. 7. Better Performance Achieved for Com-putation-intensive Applications with Changing r

(PCRAM) can provide more promising performance than hard disk drives (HDDs) [14, 15], especially for random accesses [14, 16]. However, they cannot reduce the data movement across the network. They help mitigate the performance gap between processors and I/O devices but will not be able to address the issue of large volume of data movement alone for data-intensive sciences. Active storage, active disks, and smart disks have gained increasing attention in recent years [17, 18]. Active storage leverages the computing capability of storage nodes and performs certain computation to reduce the bandwidth requirement between storage and compute nodes. Active disks and smart disks integrate a processing unit within disk storage devices and offload computations to embedded processing unit. However, these architecture improvements are designed to explore either the idle computing power of storage nodes or an embedded processor, and have limited computation-offloading capability. Blue Gene Active Storage [19] is a recent implementation of active storage prototype on the IBM Blue Gene platform. The Oracle Exadata [20] is another active storage-like system but focuses on scan-intensive database queries (read operations). The decoupled HPC system architecture provides a more powerful platform for the same purpose [21]. In addition, it handles both read and write operations. I/O forwarding (both hardware and software solutions) [22, 23] and data shipping [24] provide approaches to offloading I/O requests to dedicated nodes, aggregating the requests, and carrying out them on behalf of compute nodes. The data nodes in the decoupled system architecture can carry all these functions.

Current parallel programming models are designed for computation-intensive applications. These programming models include Message Passing Interface (MPI), Global Arrays, OpenMP, Unified Parallel C, Chapel, X10, Co-array Fortran, and data parallel programming models such as High Performance Fortran (HPF). These programming models primarily focus on the memory abstractions and communication mechanism among processes. I/O is treated as a peripheral activity and often a separate phase in these programming models, which is often achieved through a subset of interfaces such as MPI-IO [25]. Advanced I/O libraries, such as Hierarchical Data Format (HDF), Parallel netCDF (PnetCDF), and Adaptable IO System (ADIOS) [26], provide high-level abstractions, map the abstractions onto I/O in one way or another [27, 28], and complement parallel programming models in managing data access activities. The MapReduce programming model [10, 29, 30] is an instant hit and has been proven effective for many data-intensive applications. The MapReduce model, however, is typically layered on top of distributed file systems and is not designed for HPC semantics. It requires specific Map and Reduce abstractions as well [10, 29]. The decoupled HPC system architecture is studied for general HPC applications.

There have been significant amount of research efforts in optimizing data-access performance using runtime libraries. Abbasi et. al. proposed a DataStager framework with data staging services that move output data to dedicated staging or I/O nodes prior to storage, which has been proven effective

in reducing the I/O overheads and interferences on compute nodes [31]. Zheng et. al. proposed a preparatory data analytics approach to preparing and characterizing scientific data when generated (e.g. data reorganization and metadata annotation) to speedup subsequent data access [32]. These approaches have shown considerable performance improvement with dedicated output staging services and preparatory analysis. A decoupled HPC system architecture studied in this research leverages dedicated nodes as well. These data nodes can provide buffering or staging too, but more importantly on data reduction. The notion of data processing nodes is a revisit of HPC system architecture to provide balanced computational and data-access capability. The decoupled HPC system architecture considers to address the fundamental architecture issue for data-intensive sciences.

V. Conclusion

Many scientific computing applications have become increasingly data-intensive. These applications have brought up an important question to the HPC research and development community - how to efficiently support data-intensive sciences with HPC systems, while conventional HPC systems are designed for computation-intensive applications. The massive amount of data movement and long access delay can significantly limit the productivity of data-intensive scientific applications.

In this paper, we present our research study trying to answer the above question with revisiting HPC system architecture. We study a decoupled HPC system architecture for scientific big data applications. The decoupled architecture builds separate data processing nodes and compute nodes, with computation-intensive and data-intensive operations mapped to compute nodes and data processing nodes respectively. The data processing nodes and compute nodes collectively provide a balanced system design for data-intensive applications. We have presented modeling and analyses to study the potential. The result has shown a promising potential of such a decoupled HPC system architecture. We were able to draw important conclusions for HPC system design and development, and these conclusions can guide the configuration and deployment of future HPC systems for solving data-intensive scientific problems. While this study is one of steps of we are trying to develop better HPC solutions for scientific big data applications, the current results are encouraging. Given the growing importance of supporting data-intensive sciences, such a decoupled HPC system architecture can have an impact. It can potentially guide building exascale HPC systems as well to better support data-intensive sciences.

VI. Acknowledgment

This research is sponsored in part by the National Science Foundation under grant CNS-1162540, CNS-1162488, CNS-1161507, and CNS-1338078. The authors acknowledge the High Performance Computing Center (HPCC) at Texas Tech University at Lubbock for providing HPC resources that have contributed to the research results reported within this paper. URL: http://www.hpcc.ttu.edu. The authors would also like to acknowledge anonymous reviewers for their suggestions that improved this research study.

References

[1] J. Dongarra, P. H. Beckman, and T. M. etc., "The International Exascale Software Project Roadmap," *IJHPCA*, vol. 25, no. 1, pp. 3–60, 2011.

[2] V. Sarkar, S. Amarasinghe, and D. C. etc., "ExaScale Software Study : Software Challenges in Extreme Scale Systems," *ExaScale Computing Study*, pp. 1–159, 2009.

[3] "DOE Innovative and Novel Computational Impact on Theory and Experiment Program," http://hpc.science.doe.gov/.

[4] R. Ross, R. Latham, M. Unangst, and B. Welch, "Paralell I/O in Practice," in *Tutorial in the ACM/IEEE SC'09 Conference*, 2009.

[5] "Global Cloud Resolving Model (GCRM)," https://svn.pnl.gov/gcrm.

[6] V. R. Borkar, M. J. Carey, and C. Li, "Big data platforms: what's next?" *ACM Crossroads*, vol. 19, no. 1, pp. 44–49, 2012.

[7] L. Liu, "Computing infrastructure for big data processing," *Frontiers of Computer Science*, vol. 7, no. 2, pp. 165–170, 2013.

[8] T. Condie, P. Mineiro, N. Polyzotis, and M. Weimer, "Machine learning for big data," in *SIGMOD Conference*, 2013, pp. 939–942.

[9] "Community Earth System Model," http://www.cesm.ucar.edu.

[10] J. Dean and S. Ghemawat, "MapReduce: Simplied Data Processing on Large Clusters," in *Operating Systems Design and Implementation*, 2004, pp. 137–150.

[11] "IOSIG Project," http://www.cs.iit.edu/~scs/iosig.

[12] J. He, J. Bent, A. Torres, G. Grider, G. A. Gibson, C. Maltzahn, and X.-H. Sun, "I/O Acceleration with Pattern Detection," in *HPDC*, 2013, pp. 25–36.

[13] Y. Chen, S. Byna, X.-H. Sun, R. Thakur, and W. Gropp, "Hiding I/O Latency with Pre-execution Prefetching for Parallel Applications," in *Proc. of the 2008 ACM/IEEE conference on Supercomputing*, ser. SC '08, 2008, pp. 40:1–40:10.

[14] F. Chen, D. A. Koufaty, and X. Zhang, "Hystor: Making the Best Use of Solid State Drives in High Performance Storage Systems," in *ICS*, 2011, pp. 22–32.

[15] S. Chen, P. B. Gibbons, and S. Nath, "Rethinking database algorithms for phase change memory," in *CIDR*, 2011, pp. 21–31.

[16] X. Dong and Y. Xie, "AdaMS: Adaptive MLC/SLC Phase-change Memory Design for File Storage," in *ASP-DAC*, 2011, pp. 31–36.

[17] Y. Xie, K.-K. Muniswamy-Reddy, and D. F. etc., "Design and Evaluation of Oasis: An Active Storage Framework based on T10 OSD Standard," in *MSST*, 2011.

[18] S. W. Son, S. Lang, and P. e. Carns, "Enabling Active Storage on Parallel I/O Software Stacks," in *Proceedings of the 2010 IEEE 26th Symposium on Mass Storage Systems and Technologies (MSST)*, ser. MSST '10, 2010, pp. 1–12.

[19] B. G. Fitch, A. Rayshubskiy, M. P. T.J. Chris Ward, B. Metzler, H. J. Schick, B. Krill, P. Morjan, and R. S. Germain, "Blue Gene Active Storage," in *HEC FSIO R&D Workshop '10*, 2010.

[20] "Oracle Exadata Database Machine," http://www.oracle.com/us/products/database/exadata/database-machine-x3-8/overview.

[21] Y. Chen, C. Chen, X.-H. Sun, W. D. Gropp, and R. Thakur, "A Decoupled Execution Paradigm for Data-Intensive High-End Computing," in *In the Proc. of the IEEE International Conference on Cluster Computing 2012 (Cluster'12)*, 2012.

[22] N. Ali, P. H. Carns, and K. I. etc., "Scalable I/O Forwarding Framework for High-performance Computing Systems," in *CLUSTER*, 2009.

[23] K. Iskra, J. W. Romein, K. Yoshii, and P. Beckman, "ZOID: I/O-forwarding Infrastructure for Petascale Architectures," in *PPoPP*, 2008.

[24] F. Schmuck and R. Haskin, "GPFS: A Shared-Disk File System for Large Computing Clusters," in *Proceedings of the 1st USENIX Conference on File and Storage Technologies*, 2002.

[25] R. Thakur, R. Ross, E. Lusk, and W. Gropp, "Users Guide for ROMIO: A High-Performance, Portable MPI-IO Implementation," *Mathematics and Computer Science Division*, 1997.

[26] J. Lofstead, M. Polte, G. Gibson, S. Klasky, K. Schwan, R. Oldfield, M. Wolf, and Q. Liu, "Six Degrees of Scientific Data: Reading Patterns for Extreme Scale Science IO," in *Proc. of HPDC*, 2011, pp. 49–60.

[27] H. Abbasi, G. Eisenhauer, M. Wolf, K. Schwan, and S. Klasky, "Just in time: Adding Value to the IO Pipelines of High Performance Applications with JITStaging," in *HPDC*, 2011, pp. 27–36.

[28] P. Widener, M. Wolf, H. Abbasi, S. Mcmanus, M. Payne, M. Barrick, J. Pulikottil, P. Bridges, and K. Schwan, "Exploiting Latent I/O Asynchrony in Petascale Science Applications," *IJHPCA*, vol. 25, pp. 161–179, 2011.

[29] S. Sehrish, G. Mackey, J. Wang, and J. Bent, "MRAP: a novel MapReduce-based framework to support HPC analytics applications with access patterns," in *IEEE International Symposium on High Performance Distributed Computing*, 2010, pp. 107–118.

[30] R. Grover and M. J. Carey, "Extending map-reduce for efficient predicate-based sampling," in *ICDE*, 2012, pp. 486–497.

[31] H. Abbasi, M. Wolf, and G. e. Eisenhauer, "DataStager: Scalable Data Staging Services for Petascale Applications," in *HPDC*, 2009.

[32] F. Zheng, H. Abbasi, and C. D. etc., "Predata - preparatory data analytics on peta-scale machines," in *IPDPS*, 2010, pp. 1–12.

A Named Data Network Approach to Energy Efficiency in IoT

Oliver Hahm, Inria
Emmanuel Baccelli, Inria
Thomas C. Schmidt, HAW
Matthias Wählisch, FU Berlin
Cédric Adjih, Inria

***Abstract*—In the IoT, the trade-off between content availability and energy efficiency plays a crucial role. In this paper, we propose an energy-saving approach leveraging distributed caching for IoT content, based on an information-centric networking paradigm. We extend the NDN protocol with a variety of caching and replacement strategies, and we discuss alternative approaches for extending NDN to accommodate such IoT use cases. Based on experiments on real IoT hardware in a network gathering hundreds of nodes, we demonstrate these caching strategies can bring 90% reduction in energy consumption while maintaining IoT content availability above 90%.**

I. Introduction

In the IoT, energy efficiency and memory efficiency play crucial roles. The standard approach to energy efficiency in the IoT consists in combining the techniques below:

- Energy efficient hardware with micro-controller and radio consuming energy in mW range and ultra-efficient sleep modes in nW range (energy harvesting techniques may also be applicable in some cases, but are not the focus of this paper).
- Low-power radio and MAC layers based on radio duty-cycling, aiming to reduce idle listening as much as possible, e.g. ContikiMAC [1].
- Network layer protocols that are less chatty e.g. 6loWPAN stack protocols adapting IPv6 to IoT [2]
- Content caching centralized in the cloud or on a proxy, e.g. CoAP / HTTP caching [3].

In particular, with this combined approach, there is no trade-off between content availability and energy efficiency because, while IoT devices sleep a large part of the time, content availability is preserved by a proxy (or the cloud) which caches the content. With this approach, sensors can fully benefit from radio duty-cycling mechanisms allowing less than 1% radio activity [4], which can achieve 100-fold less transceiver energy consumption in some scenarios.

However, in other IoT use cases, a designated gateway/proxy is unavailable most of the time. One example is *in-the-wild monitoring of plants, soils or animals*, which requires a large number of small IoT devices embarking sensors, disseminated in an area, e.g. on a meadow. Another example is *monitoring large storage location*, which requires a large number of scattered IoT devices with sensors tracking the state of monitored goods or machines. In such use cases, there is no designated proxy and the cloud is reachable infrequently, e.g. only when a drone with appropriate communication capabilities flies by, or when an employee tours to check the area, carrying a tablet that polls sensors via radio communications. One might consider installing a designated gateway which can communicate locally with the IoT devices using a low-power radio interface, and with a high-power radio interface i.e. the uplink. However, prior work in the field of wireless sensor networks indicates that the gateway will typically also sleep a large part of the time to save energy and increase life-time. For example, in [5] authors show how wireless sensor networks can benefit from an 8-fold improvement in energy efficiency using such techniques combined with dynamic clustering.

In the above IoT use-cases, while radio duty cycling is still possible the standard approach combining sleeping and centralized caching is not applicable: a trade-off reappears between data availability and energy efficiency. In absence of a designated network element to centralize caching, an alternative approach consists in aiming for a dynamically distributed cache. The focus of this paper is thus to study mechanisms that dynamically distribute cached IoT content and allow IoT devices to be in sleep mode as often as possible, while maintaining acceptable levels of availability for IoT content.

A. Related Work

Early work in the field of wireless sensor networks proposed on-path caching [6], [7] to reduce the need for end-to-end retransmissions at the transport layer in multi-hop wireless sensor networks (WSN). In [8], authors proposed a content-aware diffusion mechanism for WSN leveraging on-path caching. A similar approach is recently the focus of a growing community: the information-centric networking paradigm (for example NDN [9]) proposes communication that is not host-centric and conversational such as with TCP/IP, but content-centric and completely connectionless. With NDN, a data consumer requests content via *Interest* packets forwarded in the network, which eventually hit either a cache containing a chunk of this content, or the content producer itself. Either way, content chunks are sent back to the consumer via reverse path forwarding, following the trail of transient Pending Interest Table (PIT) entries maintained in each intermediate node. Such nodes may then opportunistically cache such content chunks in their Content Store (CS). While initially proposed for a wired, core Internet context and large non-transient content, information-centric networking (ICN) is most recently being considered for the IoT and small

978-1-5090-2482-7/16/$31.00 ©2016 IEEE

transient content. In [10], authors experimented with NDN on an IoT testbed and hinted at potential memory- and energy-efficiency gains with such an approach compared to the traditional 6LoWPAN approach, but stopped short of studying caching and replacement strategies. In [11], a new NDN communication pattern for IoT (and an optimisation) is proposed to exploit the wireless broadcast nature of IoT networks to retrieve content from multiple producers with a single interest. Persistent PIT entries are used. In [12] authors propose complementary mechanisms to adapt NDN to information freshness requirements specific to IoT sensor data, by introducing mechanisms for consumers to express them. In [13], authors give a high-level overview of advantages, trade-offs and challenges of information-centric networking for IoT. With the purpose of tracking content changes, authors in [14] propose ChronoSync, a mechanism built on top of NDN architecture to efficiently synchronize data and data updates among multiple users (e.g. for distributed chat, distributed file synchronization). The closest related work is [15] which includes a study of a basic random caching strategy with LRU and observe performance gains in content delivery, via simulations on a grid topology.

In general, NDN can be used over various transports including IP or TCP. However, the typical constraints of IoT systems in terms of memory constraints and link layer packet size limitations, demand a deployment directly on top of the link layer [10]. Hence, while other services on top of an IP-based IoT stack could offer similar mechanisms as the ones described in the paper, in order to reduce energy consumption while maintaining the content availability, a NDN approach implements these mechanisms directly as part of the network layer and is therefore much more applicable for highly constrained IoT scenarios.

To the best of our knowledge, there is no prior work on advanced distributed caching strategies in IoT on real hardware, that addresses the trade-off of data availability and energy efficiency. In this paper, we focus on this problem. The contributions of this paper are the following:

- We extend NDN with mechanisms allowing various caching and replacement strategies combined with deep sleep strategies,
- We comparatively evaluate several caching and replacement strategies based on extensive experiments on real IoT hardware in a network gathering hundreds of nodes
- We demonstrate experimentally on real IoT hardware that NDN enhanced with these sleeping, caching and replacement strategies can bring 90% reduction in energy consumption, while maintaining the availability of recent IoT content above 90%.
- We discuss the impact of specific IoT scenarios on NDN applicability, and alternatives for NDN extension catering for such scenarios.

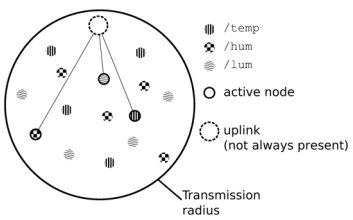


Figure 1. Broadcast domain.

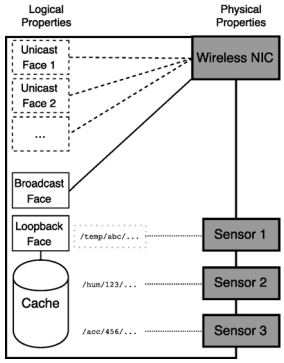


Figure 2. IoT device.

II. IoT Scenario

We base the following discussions on a basic IoT scenario: a single wireless broadcast domain that gathers a set of sensors of various types as shown in Fig. 1. This domain is connected to the Internet via an intermittent uplink.

In detail, each sensor is a content producer, and is hosted on an IoT device as shown in Fig. 2. Each IoT device provides a small cache (RAM $\approx$ 50 kB) and a low-power CPU, to which are connected peripherals including a low-power radio interface. Such an IoT device is thereafter called a *node*, and a node can host one or more sensors.

Nodes alternate between *active* and *sleeping* phase according to a sleeping strategy, which can be either coordinated or uncoordinated. Upon generation of new content, a sensor can wake up the node it is hosted on (functionality available IoT operating systems, e.g. RIOT [16]), and can push this content to replicate it in the caches of other nodes that are currently in active phase. Nodes in active phase cache new content in their content store, the details depend on caching and replacement strategies (see Section III). When the uplink turns up, all active nodes transmit their cached data to the uplink. We focus on scenarios where sensors monitor a phenomenon whereby (i) data relevance strictly decreases with time, and (ii) a more complete view of what the sensors are monitoring is achieved if available data comes from a larger number of distinct sources (i.e. sensors).

In the subsequent sections, we discuss two crucial aspects in this context, the sleeping strategy and the cache replacement strategy.

III. Design Space: Strategies for Sleeping, Caching & Cache Replacement in NDN IoT

Our goal of deploying NDN in the IoT is to improve energy efficiency while maintaining availability of recent data. The core questions that need to be addressed are (i) how to organize sleeping of nodes to best use scarce energy resources, and (ii) how to organize cache maintenance as memory is limited?

We define and analyze a number of approaches for sleeping and caching. We distinguish between uncoordinated and coordinated approaches. In the **uncoordinated approach** nodes sleep in a randomized manner: each

node sleeps with probability p for a given period of time. Sleeping nodes rely on the fact that p is chosen in a way that the expectation for active nodes is ≥ 1 at any given point of time to receive and cache their data on their behalf. In the **coordinated approach**, a deterministic mechanism assigns only one node (per prefix and per period of time) which is then active and responsible for receiving and caching the data of sleeping nodes.

A. Uncoordinated Sleeping, Random Caching

In completely uncoordinated environments we cannot assume an administrative authority which pre-configures nodes. Each node thus decides every xD seconds whether it will be active or sleeping for the next xD seconds. The probability for sleeping mode is given by the parameter p. Active nodes that receive a content chunk will try to cache it with probability $q = 0.5$, similar to the caching approach depicted in [15]. The cache replacement strategy is a First-In First-Out (FIFO) policy, roughly equivalent to Least Recently Used (LRU) in this context.

B. Coordinated Sleeping

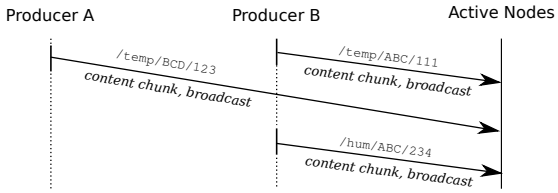
Single Deputy. This approach leverages coordination of nodes' sleeping phases. During the network's bootstrap, nodes determine an absolute order between them. The node in the first position of this order is elected as the first *deputy* and stays active for a certain period. Based on the determined order, each node will successively become *deputy* following a round-robin scheme. When a node wakes up to become the next deputy, it takes over deputy role by requesting the full cache from the previous deputy (using the simple Interest-based mechanism used by the uplink to request all available content from active nodes).

Multiple Deputies. In scenarios where the amount of relevant content exceeds a single cache, a single deputy is not sufficient. Hence, we introduce multiple deputies responsible for different prefixes like */hum*. This can be pre-configured by the network operator.

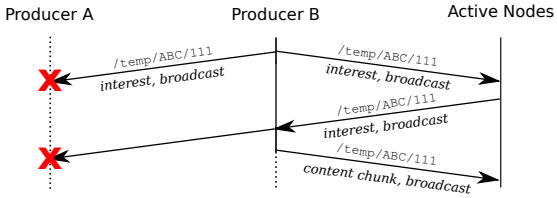
C. Name-based Cache Replacement

Max Diversity Most Recent (MDMR). To implement a cache that maximizes diversity of content wrt of different sensor sources, the IoT may benefit from the naming scheme in NDN. New content name is derived locally using the type of sensor (prefix) and the timestamp (suffix) see details in section V. The cache replacement strategy works then as follows: First, the cache tries to replace older chunks from the same producer. Next, the cache tries to replace the oldest chunk of a producer from which several chunks are present in the cache. Finally, if there is only entry per source, the oldest entry in the cache is replaced.

Prioritized Prefixes (P-MDMR). This prefix can be autoconfigured using local information coming from the (main) sensor of the node. E.g., a node with a temperature sensor will prioritize content for prefix */temp*. A node always tries to cache content chunks for the prioritized

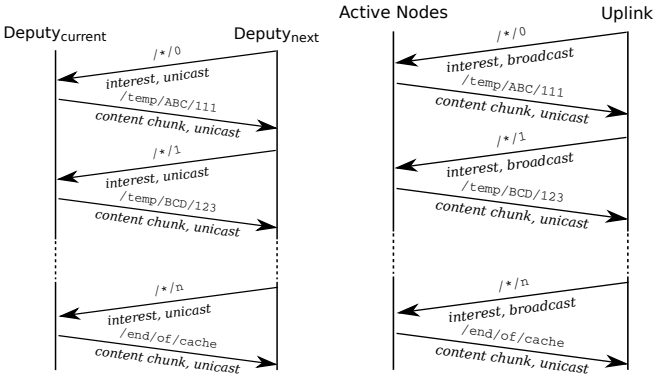


(a) Distributing data from sensors to cache as unsolicited content chunk.



(b) Distributing data from sensors to cache using Interest-Interest.

Figure 3. Basic communication schemes between content producers and caches.



(a) Distributing data between two caches.

(b) Distributing data from cache to uplink.

Figure 4. Basic communication schemes among deputies and between deputies and uplink.

prefix, while other content are cached with a probability $q < 1.0$. Fig. 1 depicts a network with three different types of sensors and consequently three different prioritized prefixes scattered in the local IoT network. If the cache is full, entries for non-prioritized prefixes are replaced first.

D. Basic Implementation Requirements

To implement the mechanisms described above in common IoT scenarios (see Section II), little changes are necessary based on NDN:

Opportunistic Caching of Unsolicited Content. When producers wake up, they want to offload content and immediately go back to sleep in order to save energy. The most basic approach is the distribution of new content via broadcast. In consequence, active nodes need to accept such broadcasts and allow for opportunistic caching of pushed, unsolicited content.

Several approaches for that are possible. For the experiments, we used both *Interest-Interest* [19] and an Immediate Broadcast (*IB*) approach similar to [20], which allows *producers* to immediately push their new content (active nodes have a permanent *PIT* entry matching the wildcard

prefix /∗ and hence, do not discard unsolicited content chunks). Note, that although *IB* does not comply with the "self-regulation of traffic" principle of NDN, it does not harm the flow-balance of the network, since it only occurs between two nodes and is never forwarded. We observed that, the main advantages of *Interest-Interest* compared to *IB* is that the Interest handshake conforms more easily with basic NDN, and increase robustness in face of packet loss typically experienced on IoT link layers, when this link layer is not too congested. The main drawback of *Interest-Interest* compared to *IB* is that it incurs significantly more control traffic and can thus be both (i) less energy efficient and (ii) counter-productive in terms of packet loss if the link layer is already congested. In particular, *Interest-Interest* cannot be used with *uncoord. sleeping* because (potentially) too many active nodes answer the initial Interest. Instead, preventive retransmissions (e.g. broadcasting 3 times upfront) can improve robustness against potential packet loss.

Interest Signaling for Group of Content. When a new deputy wakes up, or when an uplink appears, cached content should be transferred. However, nodes may not be aware of previously distributed content and therefore cannot request each content chunk explicitly. To request data for an unknown name, we require a wildcard symbol (e.g., `<prefix>/*`), which expresses interest for all content under the prefix.

Having those mechanisms in place, nodes can locate and transfer content without relying on strictly synchronized schemes or significant prior knowledge. We summarize the traffic exchanges between producers and active nodes in Fig. 3, and between active nodes and uplink in Fig. 4. The next deputy respectively the uplink request the whole CS from the current deputy chunk by chunk using the wildcard symbol. In the coord. sleeping approach a dedicated link layer address can be assigned to the current deputy, allowing for a unicast Interest. The uplink however must always use broadcast for its Interests.

IV. Experimental Evaluation

A. Content Availability Metrics

The goal of this paper is to study mechanisms that allow high sleeping/active ratio (p) while tending towards, or achieving atomic availability. Basically, this means that when the uplink requests content, recent content is available as if all nodes were in active state. We evaluate the availability of *recent* IoT content based on two metrics.
Diversity Metric:

$$Diversity := \frac{|C_{disj}|}{|S|} \tag{1}$$

where $|S|$ is the number of content producers (i.e. sensors) in the network, and $|C_{disj}|$ is the number of disjoint name prefixes present in the aggregated cache C_{AGG} of all the currently active nodes.

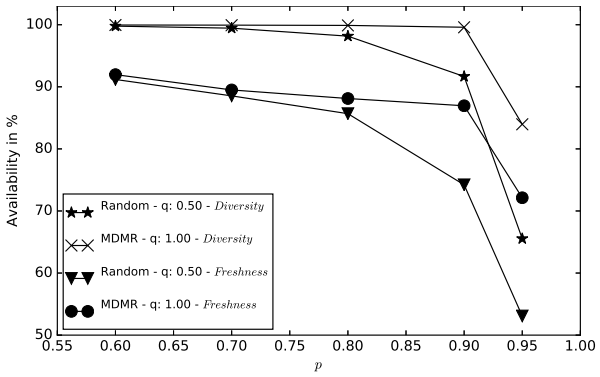


Figure 5. Availability for *uncoord. sleeping* approaches with different values for p wrt $Diversity$ and $Freshness$. Experiments were conducted with 50 nodes.

Tolerated Freshness Metric:

$$Freshness := \frac{\sum_{i=1}^{|C_{disj}|} (c_i | AGE(c_i) < tD)}{|S|} \tag{2}$$

where c_i is an entry in C_{AGG}, $AGE(c_i)$ its age relatively to the newest entry from the same source, and t is a threshold value defining "tolerably" recent content. For the rest of the paper we will use the terms $Diversity$ and $Freshness$ to refer to the above defined metrics.

B. Implementation

Implementation with RIOT and CCN-lite. We implemented the caching, replacement, and sleeping strategies on top of RIOT [16] [1]. RIOT supports the NDN implementation of CCN-lite [17] as a *package*. We used some hooks in the CCN-lite protocol engine to implement small modifications to the processing of Interests and content chunks. For instance, a mechanism similar to the `mod_rewrite` module on HTTP daemons was introduced to rewrite Interests for $/*/N$ on active nodes to match the Nth entry in the content store.[2]

Experimental Setup on FIT IoT-LAB. The experiments were conducted on ≈ 220 nodes deployed over a 225 m^2, which are part of the Lille site of the FIT IoT-Lab testbed [18]. Each run lasted for 20 minutes, D was set to 3 s and xD to 12 s. Nodes were of the *M3* type, which are equipped with an 32-bit ARM Cortex-M3 MCU, 64 kB of RAM, 256 kB of ROM, an IEEE 802.15.4 2.4 GHz radio transceiver and four different sensors (light, accelerometer, gyroscope, pressure). In order to evaluate the energy consumption for these experiments, we measured the duration that a node spent in active and sleeping state plus the number of unicast and broadcast transmissions. Based on these hardware and network characteristics, we derive a simple energy consumption model: $E = \sum_{state} P_{state} \cdot t_{state}$, where P_{state} defines the power consumed for a given *state* and t_{state} is the time to spent in this state. We define the states: *sleeping*, *active* (listening

[1] Source Code is available at `http://ndnrg.riot-os.org/ccnl_caching`
[2] Chunks in the *CS* of each node have a arbitrary, but fixed order.

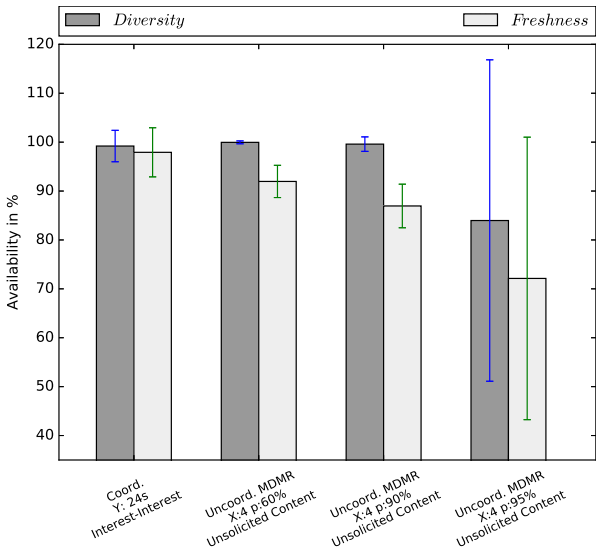


Figure 6. Comparing Availability wrt *Diversity* and *Freshness* between *coord.* and *uncoord. sleeping* approaches. Experiments were conducted with 50 nodes, Y represents the *deputy* cycle interval.

and receiving), *sending unicast*, and *sending broadcast* packets. Values for power consumption per state are taken from the datasheets of the MCU and radio transceiver. Furthermore, we assume a typical RDC rate of 0.6%, i.e. the default value for ContikiMAC.

C. Experiment Results

We evaluate two different cases: (i) the case where the cache of single node (size C) is bigger than the number of producers ($|S|$), and thus a single active node could cache all the recent content, and (ii) the case where C is smaller than $|S|$, i.e. multiple nodes must be active to cache all the recent content. In the second case, p must be chosen in a way, so that the expectation $(1-p)|N| \cdot C \geq |S|$.

Small number of producers. First, we consider the case where $C \geq |S|$. We compare how *uncoord. sleeping* approaches perform for different values of p. We analyze only availability here, since an analysis of the energy consumption does not reveal any surprises: the time spent in active mode can be derived from p and the number of sent packets do not differ for the different approaches. It is worth noting that sensors hosted on currently active nodes also broadcast their newly generated data.

In Fig. 5 we notice surprising performance until a sharp drop in availability beyond $p = 0.9$. In detail, the *Random Caching* strategy achieves good diversity, with values just below 100% *Diversity* with $p \leq 0.9$, but drops below 70% above that threshold, e.g. for $p = 0.95$. Comparatively, the *Name-based Caching* strategy consistently achieves 100% for $p \leq 0.9$ and still a decent value of $\approx 85\%$ for $p = 0.95$.

Furthermore, regarding the *Freshness* metric (see Fig. 5), we notice that availability also drops for $p \geq 0.9$, below 80% for *Name-based Caching*, and much less, around 50%, for *Random Caching*.

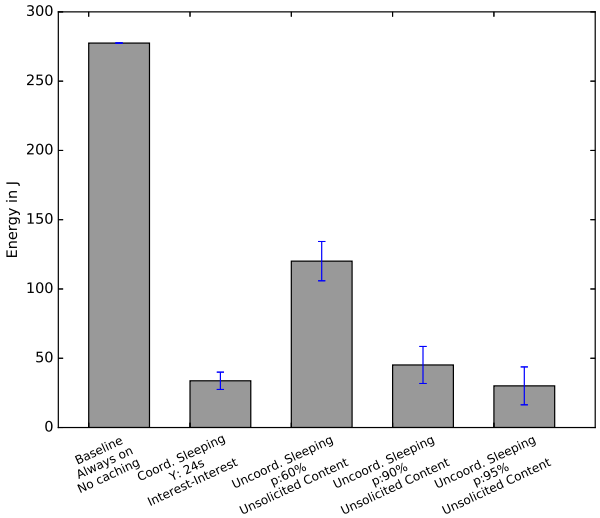


Figure 7. Comparing average energy consumption per node for *coord.* and *uncoord. sleeping* approaches.

Now let's compare these results with the performance of the *coord. sleeping*. Fig. 6 reveals that, compared to the *uncoord. sleeping* approach with $p \leq 0.9$, the *coord. sleeping* approach achieves similar diversity. However, in terms of *Freshness*, the *coord. sleeping* approach consistently outperforms *uncoord. sleeping* even when p is small.

In order to evaluate the energy consumption for these experiments, we measured the duration each node spends in active and sleeping state, and the number of unicast and broadcast transmissions. We then fed these values to the energy consumption model derived in Section IV-B. We compare energy consumption to a baseline where each node caches only the data it produces, has its CPU always on, but use state-of-the art radio duty-cycling with ContikiMAC. (Note that, hence, for the baseline, there is no communication between the nodes, only with the uplink). The results are shown in Fig. 7. With *coord. sleeping*, more network traffic is induced, but nodes can spend longer time in sleep mode, which compensates the energy consumption of the communication overhead. We observe that we can reduce the energy consumption by about 90% compared to the baseline without affecting the data availability (compare Fig. 6).

Large number of producers. Now, we analyze the content availability for the *uncoord. sleeping* approaches for scenarios where $|S| > C$. Fig. 8 presents these results for *Random Caching* and *Name-based Caching* strategies. We observe that availability (both *Diversity* and *Freshness*) decreases drastically with *Random Caching* and basic *Name-based Caching* strategies. To improve the availability, we use the P-variant leveraging *prioritized prefixes* to better use the capacity of the aggregated cache C_{AGG}. In detail, nodes will now cache content for an autoconfigured priority prefix with a probability of 1.0, and other content with probability 0.5. Using P-MDMR with 3 different types of sensors (i.e. 3 prioritized prefixes) availability is

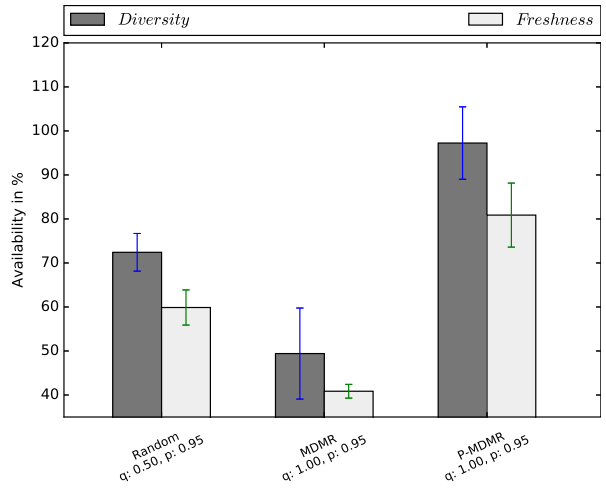


Figure 8. Availability wrt *Diversity* and *Freshness* for $|S| > |C|$ comparing different mechanisms for the *uncoord. sleeping* approaches. Experiments were conducted with 220 nodes and $p = 0.95$.

substantially better, both in terms of diversity (back above 90%), and in terms of freshness ($\approx$ 80%).

V. Discussion

Potential Energy Savings via Link Layer Name Mapping. Further energy savings may be achieved if *appropriate mapping of names to link layer* information could be leveraged. For example, one could map a specific name prefix to a specific PAN ID, and based on prefix priority, filter out radio packets in hardware, without having to wake up the CPU. This would be the equivalent of an overheard radio unicast being discarded at transceiver level, instead of at CPU level, which can thus remain in power-save mode and save further energy.

IoT Content Names. We considered that each sensor reading value fits in a single chunk and that each chunk can fit in one radio packet. Typical MTU on an IoT network is in the 100 bytes range (e.g. 127 bytes using IEEE 802.15.4). In practice, a sensor reading value coded on 32bits offers a quite reasonable range in most cases, and leaves substantial space for names. Therefore, while ICN schemes based on short names are necessary in this context, it does not seem a major blocker. For example, in our IoT deployment (as for any large IoT deployment), manual name configuration was out of the question. To bootstrap our deployment, names had to be autoconfigured. Each name must be derivable locally and must satisfy the requirements of (i) meaningfulness, and (ii) uniqueness. In order to satisfy the first requirement, we assumed a prefix that derived from sensor type identifier and a unique identifier of the node, e.g. a vendor ID. To fulfill the second requirement, we extend the prefix of the name by a suffix, the timestamp, which can also serve as a version number. The name could be enhanced by further information, e.g., based on geographical or organizational properties. A name generated by the autoconfiguration mechanism looks like $/hum/DEADBEEF/1466250645$.

VI. Conclusions & Future Work

In this paper, we have proposed and studied experimentally a number of mechanisms for name-based, decentralized, cooperative caching of IoT content, which allow to capture most of the phenomenons observed by IoT devices' sensors, while draining an order of magnitude less energy, compared to prior art. In practice, we adapted the NDN protocol for new IoT scenarios, and we extended NDN with novel caching and replacement strategies, which we have implemented and experimented on real hardware. Our testbed experiments show that these mechanisms achieve 90% reduction in energy consumption, while maintaining tolerably recent content availability above 90%. Our future work will focus on improving content availability even further while achieving similar energy consumption gains, when the number of nodes and the amount of new IoT content generated in the network grows larger.

References

[1] A. Dunkels, "The contikimac radio duty cycling protocol," 2011.
[2] Z. Sheng *et al.*, "A Survey on the IETF Protocol Suite for the Internet of Things: Standards, Challenges, and Opportunities," in *IEEE Wireless Communications*, 2013.
[3] Z. Shelby *et al.*, "The constrained application protocol (CoAP)," Tech. Rep., 2014.
[4] D. Stanislowski *et al.*, "Adaptive synchronization in IEEE802.15.4e networks," in *IEEE Transactions on Industrial Informatics*, 2014.
[5] W. R. Heinzelman *et al.*, "Energy-efficient communication protocol for wireless microsensor networks," in *IEEE System Sciences*, 2000.
[6] F. Stann and J. Heidemann, "RMST: Reliable data transport in sensor networks," in *Sensor Network Protocols and Applications*, 2003.
[7] A. Dunkels *et al.*, "Making tcp/ip viable for wireless sensor networks," *SICS Research Report*, 2003.
[8] C. Intanagonwiwat *et al.*, "Directed diffusion: a scalable and robust communication paradigm for sensor networks," in *ACM MobiCom*, 2000.
[9] V. Jacobson *et al.*, "Networking named content," in *ACM CoNEXT*, 2009.
[10] E. Baccelli *et al.*, "Information Centric Networking in the IoT: Experiments with NDN in the Wild," in *ACM ICN*, 2014.
[11] M. Amadeo *et al.*, "Multi-source data retrieval in IoT via named data networking," in *ACM ICN*, 2014.
[12] J. Quevedo *et al.*, "Consumer-driven information freshness approach for content centric networking," in *IEEE INFOCOM Workshop*, 2014.
[13] A. Lindgren *et al.*, "Design choices for the IoT in information-centric networks," in *CCNC*, 2016.
[14] Z. Zhu and A. Afanasyev, "Let's ChronoSync: Decentralized Dataset State Synchronization in Named Data Networking," in *IEEE ICNP*, 2013.
[15] M. A. M. Hail *et al.*, "On the Performance of Caching and Forwarding in Information-Centric Networking for the IoT," in *Wired/Wireless Internet Communications*, 2015.
[16] E. Baccelli *et al.*, "RIOT OS: Towards an os for the internet of things," in *IEEE INFOCOM Poster*, 2013.
[17] "CCN Lite: Lightweight implementation of the Content Centric Networking protocol," 2014. Available: http://ccn-lite.net
[18] C. Adjih *et al.*, "FIT IoT-LAB: A large scale open experimental IoT testbed," in *IEEE WF-IoT*, 2015.
[19] J. Burke *et al.*, "Securing instrumented environments over content-centric networking: the case of lighting control and NDN," in *IEEE INFOCOM Workshop*, 2013.
[20] M. Amadeo *et al.*, "Internet of things via named data networking: The support of push traffic," in *NOF*, 2014.

IVYSPRING INTERNATIONAL PUBLISHER

International Journal of Medical Sciences

2017; 14(2): 181-190. doi: 10.7150/ijms.17119

Research Paper

Immune Cell Repertoire and Their Mediators in Patients with Acute Myocardial Infarction or Stable Angina Pectoris

Wenwen Yan[1], Yanli Song[2], Lin Zhou[1], Jinfa Jiang[1], Fang Yang[3], Qianglin Duan[1], Lin Che[1], Yuqin Shen[1]✉, Haoming Song[1]✉, Lemin Wang[1]✉

1. Department of Cardiology, Tongji Hospital, Tongji University School of Medicine, Shanghai 200065, China;
2. Department of Emergency Medicine, Tongji Hospital, Tongji University School of Medicine, Shanghai 200065, China;
3. Department of Experimental Diagnosis, Tongji Hospital, Tongji University School of Medicine, Shanghai 200065, China.

✉ Corresponding authors: Lemin Wang, Haoming Song, Yuqin Shen, Department of Cardiology, Tongji Hospital, Tongji University School of Medicine, 389 Xincun Rd, Putuo District, Shanghai 200065, China; Tel: 86 21 66111329, Fax: 86 21 66111329, E-mail: wanglemin@tongji.edu.cn; songhao-ming@163.com; sy-1963@126.com.

Received: 2016.08.05; Accepted: 2016.12.21; Published: 2017.02.08

Abstract

Background: To evaluate the natural innate and adaptive immunity through gene expression and cytology levels in peripheral blood mononuclear cells in patients with acute myocardial infarction (AMI), stable angina pectoris (SAP) and controls.

Methods: 210 patients with AMI, 210 with SAP, and 250 clinical controls were recruited. Whole human genome microarray analysis was performed in 20 randomly chosen subjects per group were examined to detect the expressions of complement markers, natural killer cells, T cells and B cells. The quantity of these cells and related cytokines as well as immunoglobulin levels were measured in all subjects.

Results: In AMI group, the mRNA expressions of late complement component, markers of natural killer cells, CD3+, CD8+ T cells and B cells were down-regulated, while those of early complement component and CD4+T cells were up-regulated ($p<0.05$). In both AMI and SAP patients, the quantity of natural killer cells, CD3+, CD8+ T cells, B cells, IgM and IgG were significantly lower than those of the controls. CD4+ T cells, CH50, C3, C4, IL-2, IL-4, IL-6 and IFN-γ were significantly higher ($p<0.05$).

Conclusions: In AMI patients, both of gene expressions related to complement, natural killer cells, CD3+, CD8+ T cells, B cells and the quantity of these immune cells decreased while cell number reduced in SAP patients. Immune function in both AMI and SAP patients decreased especially in AMI patients with declined gene and protein levels. To improve the immune system is a potential target for medical interventions and prevention in AMI.

Key words: myocardial infarction, stable angina pectoris, gene expression, innate immunity, adaptive immunity.

Introduction

Cardiovascular diseases (CADs), with high morbidity and mortality worldwide, are caused mainly by atherosclerosis. In particular, acute myocardial infarction (AMI) represents life-threatening conditions during the history of CAD [1, 2]. Nowadays we are still unable to effectively predict and prevent AMI occurrence. The pathologic mechanism responsible for majority of AMI is the rupture of stable atherosclerotic plaque and thrombosis [3]. Obviously, there must be a trigger to induce the sudden rupture. Infection seems to be undoubtedly linked to vulnerable atherosclerotic

lesion; however, its role cannot be easily documented [4, 5, 6]. Various exogenous microorganism infections, including Chlamydia pneumoniae, Helicobacter pylori, Cytomegalovirus and Bacteroides gingivalis are accepted as the new susceptible factors of CAD [7, 8].

Our recent study demonstrated the decreased T cell immunity function in AMI patients [9, 10]. T cells, as a key component of adaptive immune system, eliminate the pathogenic microorganisms and malignant cells. The significant decline of T cell function suggests that the pathogenesis of acute thrombosis in AMI patients may be associated with the depletion of immune cells. However, less is known about the nature of immune response in different stages of CAD [11, 12]. In recent study, we designed this *in vitro* study to investigate both innate and adaptive immunity in patients with AMI or stable angina pectoris (SAP). Human microarray analysis was used to systematically measure them RNA expression of the complement component, markers of immune cells in peripheral blood mononuclear cells (PBMCs) from AMI, SAP and controls. Moreover, the quantity of immune cells, related cytokines and immunoglobulin levels were also measured.

Material and Methods

Patients' Information

The study recruited 210 patients with AMI, 210 with SAP, and 250 clinically controls. Human microarray analysis was performed in 20 randomly chosen subjects per group. The sample sizes and the number of subjects per group were based on an assumed within-group variance of 0.50 and the targeted nominal power of 0.95 [13]. Table 1 shows the baseline demographic data. All patients were enrolled between Mar 2013 and Feb 2015 from our Coronary Care Unit and Cardiovascular Department. The AMI patients were admitted no more than 12 hours from the onset of symptoms to our Coronary Care Unit including 180 males and 30 females, with an average age of 59±11 years. The SAP group included 210 patients (176 males, 34 females, aged 64±11 years). 250 healthy volunteers (207 males, 43 females, aged 61± 9 years) were enrolled as the control group during the same period. Histories, physical examination, ECG, chest radiography and routine chemical analyses showed the controls had no evidence of coronary heart diseases.

All AMI patients were diagnosed on the basis of following criteria [14]: Detection of a rise of cardiac biomarker values [preferably cardiac troponin (cTn)] with at least one value above the 99th percentile upper reference limit (URL) and with at least one of the following: 1) Symptoms of ischemia. 2) New or presumed new significant ST-segment-T wave (ST-T) changes or new left bundle branch block (LBBB). 3) Development of pathological Q waves in ECG. 4) Imaging evidence of new loss of viable myocardium or new regional wall motion abnormality.5) Identification of an intracoronary thrombus by angiography.

All SAP patients had exclusively effort-related angina with a positive exercise stress test and at least one coronary stenosis was detected at angiography (>70% reduction of lumen diameter).

There were no significant differences among three groups in age, sex, body mass index (BMI), ethnicity, smoking status, systolic blood pressure (SBP), diastolic blood pressure (DBP), low-density lipoprotein cholesterol (LDL-C), triglycerides, high-density lipoprotein cholesterol (HDL-C) and fasting plasma glucose (FBG) (Table 1).

The exclusion criteria for three groups were as follows: venous thrombosis, history of severe renal or hepatic diseases, hematological disorders, acute or chronic inflammatory diseases and malignancy.

The study protocol was approved by the ethics committee of Tongji University and informed consent form was obtained.

Table 1. Baseline demographic data in three groups ($\bar{x}$ ± s.d.).

Index	AMI (a) (N=210)	SAP (b) (N=210)	Con(c) (N=250)	P (all)	P (a v b)
Age	58.5±10.7	63.6±11.1	60.9 ± 9.4	0.141	0.211
Sex(M/F)	180/30	176/34	207/43	0.694	0.773
BMI(kg/m²)	24.6±2.9	22.5±2.2	22.7±1.9	0.112	0.76
Ethnicity, Han	210	210	250	1	1
Tobacco smoking(num/d)	13.6±10.1	14.4±8.4	11.2±6.1	0.24	0.648
SBP (mmHg)	130.1±11.3	123.7±10.1	124.8±7.8	0.145	0.701
DBP (mmHg)	67.7±8.8	72.0±8.8	77.6±3.6	0.126	0.24
LDL-C(mmol/L)	2.8±1.2	2.4±1.8	2.7±1.5	0.44	0.676
Triglycerides(mmol/L)	1.5±1.8	1.7±1.0	1.8±0.7	0.51	0.12
HDL-C(mmol/L)	0.7±0.9	0.8±0.7	0.9±0.2	0.11	0.303
FBG (mmol/L)	5.3±0.4	5.1±0.7	5.0±0.2	0.24	0.834

Footnotes: BMI= body mass index; SBP=systolic blood pressure; DBP =diastolic blood pressure; LDL-C=low-density lipoprotein cholesterol; HDL-C: high-density lipoprotein cholesterol; FBG: Fasting Plasma Glucose.

Gene Expression Chips

Agilent G4112F Whole Human Genome Oligo Microarrays purchased from Agilent (USA) were used in the chip analysis. A microarray is composed of more than 41,000 genes or transcripts, including targeted 19,596 entrez gene RNAs. Sequence information used in the microarrays was derived from the latest databases of RefSeq, Goldenpath, Ensembl and Unigene [15]. More than 70% of the gene functions in the microarray are already known. All 20 randomly selected patients for each group were subjected to the chip analysis.

Total RNA Isolation

Ten milliliter of peripheral blood samples from median cubital vein were drawn from all the patients immediately after admission. Four milliliter blood was kept in PAXgene tube for total RNA isolation and the rest six milliliter was for laboratory assays. Leucocytes were obtained through density gradient centrifugation with Ficoll solution and the remaining red blood cells were destroyed by erythrocyte lysis buffer (Qiagen, Hilden, Germany). Following the manufacturer's instructions, total RNA was extracted and purified using PAXgeneTM Blood RNA kit (Cat#762174, QIAGEN, GmBH, Germany). We further checked for a RIN number to inspect RNA integration by an Agilent Bio analyzer 2100 (Agilent technologies, Santa Clara, CA, US). The sample was considered qualified when both 2100 RIN and 28S/18S were larger than or equal to 0.7.

RNA Amplification and Labeling

Total RNA was amplified and labeled by Low Input Quick Amp Labeling Kit, One-Color (Cat#5190-2305, Agilent technologies, Santa Clara, CA, US), following the manufacturer's instructions. Labeled cRNA was purified by RNeasy mini kit (Cat#74106, QIAGEN, GmBH, Germany).

Microarray Hybridization

Each slide was hybridized with 1.65ìg Cy3-labeled cRNA using Gene Expression Hybridization Kit (Cat#5188-5242, Agilent technologies, Santa Clara, CA, US) in Hybridization Oven (Cat#G2545A, Agilent technologies, Santa Clara, CA, US), following the manufacturer's instructions. After 17 hours of hybridization, slides were washed in staining dishes (Cat#121, Thermo Shandon, Waltham, MA, US) with Gene Expression Wash Buffer Kit (Cat#5188-5327, Agilent technologies, Santa Clara, CA, US), according to the manufacturer's operation manual.

Chip Scan and Data Acquisition

Slides were scanned using Agilent Microarray Scanner (Cat#G2565CA, Agilent technologies, Santa Clara, CA, US) with default settings. Dye channel: Green, Scan resolution=3ìm, 20bit. Data were extracted with Feature Extraction software 10.7 (Agilent technologies, Santa Clara, CA, US). Raw data were normalized using Quantile algorithm, Gene Spring Software 11.0 (Agilent technologies, Santa Clara, CA, US).

RT-PCR

The spots in the microarray were randomly selected and their expressions were confirmed by RT-PCR. Among all the genes with different expressions, three genes were randomly selected and subjected to RT-PCR, along with the house keeping genes (GAPDH). The relative expressions were indicated as the expression of the target genes normalized to the expression of GAPDH (2-ÄÄCt). The melting curve and the 2-ÄÄCt-method were used to detect the differences in the expressions among the three groups. The results from RT-PCR were consistent with the microarray analysis.

Laboratory assays

Two milliliter blood sample was anticoagulated with EDTA-K3 for the counting of CD16+CD56+ natural killer cells, T lymphocyte subsets and CD19+B cells, and the rest four milliliter was separated by centrifugation within 1 hour for the examination of serum immunoglobulin and cytokines. All tests were finished within two weeks.

CH50 was detected with liposome immune assay (Beckmann DxC-800 fully automatic biochemical analyzer, USA; Reagents: Wako Pure Chemical Industries, Ltd., Japan). C3 and C4 were detected with immunone-phelometry (BNII system, Siemens AG, Germany; Reagents: Siemens Healthcare Diagnostics Products GmbH, Germany).

Cytokines, including IL-2, IL-4, IL-6 and IFN-γ were measured by double antibody sandwich ELISA assay (Microplate reader Model 2010, Anthos, Austria; Reagents: Dili biotech, Shanghai). Serum levels of IgA, IgM and IgG were calculated by the immunonephelometric technique using the automated IMMAGE 800 immunochemistry system (Beckman Coulter, Brea, CA, USA), and expressed as g/L.

Leukocyte subpopulations were measured by flow cytometry (BEPICS XL-4, BECKMAN-COULTER). Monoclonal antibodies against CD3, CD4, CD8, CD16, CD56 and CD19 were purchased from BD Biosciences. The antibodies were marked with one of three fluorochromes: fluorescein isothiocyanate (FITC), phycoerythrin (PE) and phycoerythrin-cyanin 5.1 (PC5). Cells were identified by combinations as follows: CD3 (FITC)/CD16 (PE)/CD56 (PC5) (NK cells), CD3 (FITC)/CD4 (PE)/CD8 (PC5) (CD4andCD8 cells), and CD19 (PE) (B cells). In brief, 100 µL of EDTA treated blood was added to each tube and control tube was also included. 20 µL of mouse IgG1-FITC, IgG1-PE or IgG1-PC5 was then added, followed by addition of corresponding fluorescence antibodies. Following vortexing, incubation was done in dark for 30 min at room temperature. 500 µL of hemolysin (BECKMAN-COULTER) was then added, followed by incubation at 37°C for 30 min. Following washing, 500 µL of

sheath fluid was added to each tube, followed by flow cytometry (EPICS XL-4, BECKMAN-COULTER). The PMT voltage, fluorescence compensation and sensitivity of standard fluorescent microspheres (EPICS XL-4, BECKMAN-COULTER) were used to adjust the flow cytometer and a total of 10,000 cells were counted for each tube. The corresponding cell population in the scatter plot of isotype controls was used to set the gate, and the proportion of positive cells was determined in each quadrant (%). SYSTEM-II was used to process the data obtained after flow cytometry.

Statistical Analysis

Descriptive statistics were expressed as mean ± s.d. Differences between groups were examined by one-way analysis of variance (ANOVA). After ANOVA the test of all pairwise group mean comparison was performed using the Tukey's method. Density curves for CH50, C3, C4, CD16+CD56+, CD3+, CD4+, CD8+and CD19+ cells were delineated using R software. Data were analyzed using SPSS 17.0, and p-values <0.05 were considered statistically significant.

Results

Gene expression and serum level of the complement

The results showed mRNA expressions of early and late complement components including C1qá, C1qâ, C1qɤ, C1r, C1s, C2, C3, C4b, C5a, C6, C7, C8á, C8â, and C8ɤ and C9 were examined in PBMCs from three groups of patients (Figure 1A). In AMI group, gene expressions of C1qá ($p<0.05$), C1qâ, C1qɤ, C1r and C5a were significantly up-regulated (all $p<0.01$), whereas expressions of C7, C8â and C9 were significantly down-regulated when compared with SAP patients and controls, respectively ($p<0.05$). C1s expression in AMI patients was significantly lower than the controls ($p<0.05$). Serum CH50, C3 and C4 levels were significantly increased in AMI and SAP patients when compared with controls ($p<0.01$). CH50 was significantly higher in AMI patients than in SAP patients ($p<0.01$). There was no significant difference between AMI and SAP patients in C3 and C4 levels. The density curves of CH50, C3 and C4 are shown in Figure 1B-D separately.

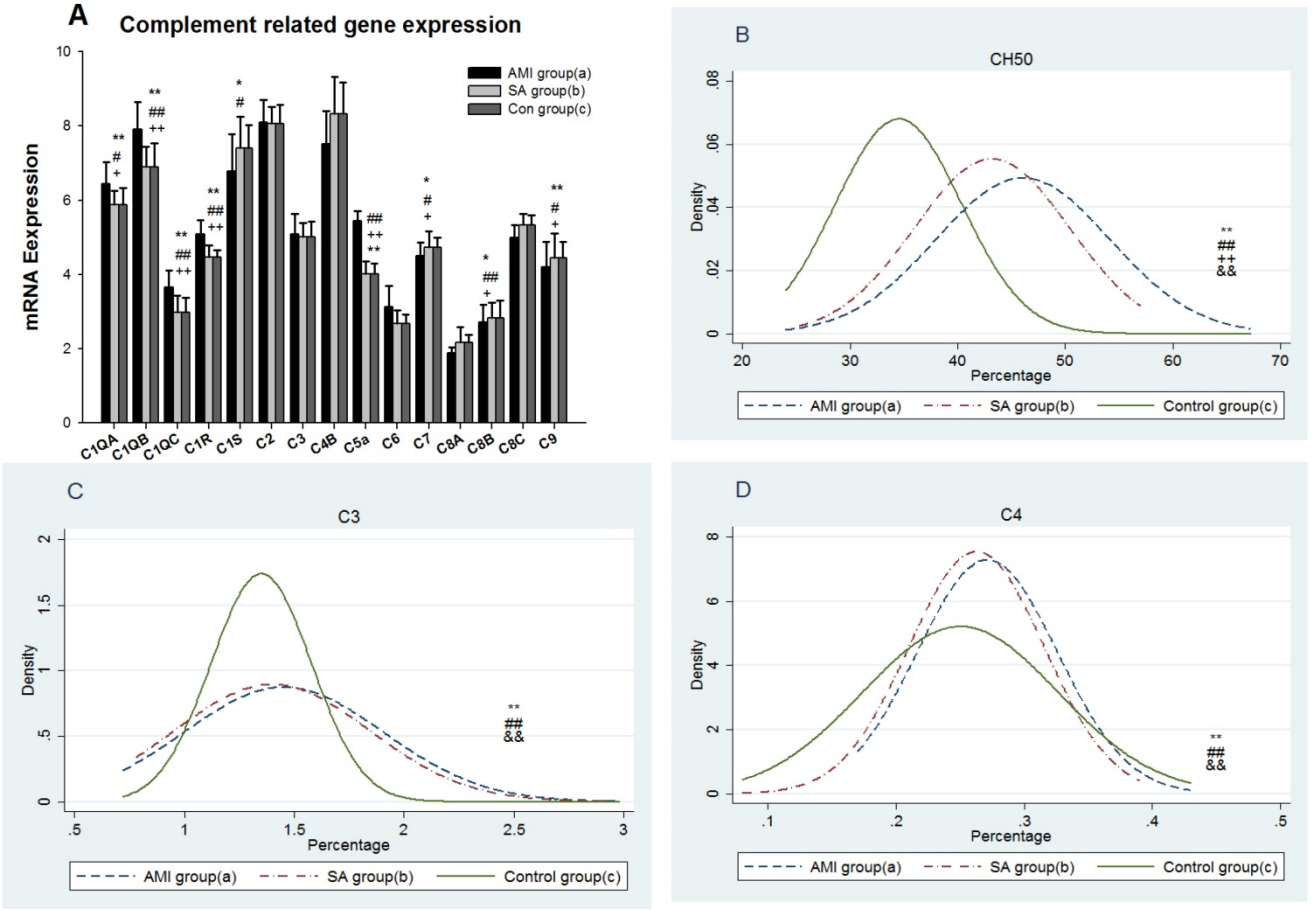


Figure 1. From three groups in PBMCs, (A) mRNA expression of early and late components complement. (B) Serum CH50 level. (C) Serum C3 level. (D) Serum C4 level. Three groups: *, P<0.05; **, P<0.01. AMI vs. Con: #, P<0.05; ##, P<0.01. AMI vs. SAP: +, P<0.05; ++, P<0.01. SAP vs. Con &: P<0.05; &&: P<0.01.

Gene expression and counting of NK cells

The results showed 12 gene expressions of NK cell biomarkers[16], including CD16, CD56, five inhibitory receptors,CD94, NKG2A, CD158(KIR2DL), CD161(KLRB1), CD328(Siglect-7) and five activating NK cell receptors, including CD335(Nkp46), CD337(Nkp30), CD48(2B4), CD314(NKG2D) and CD319(CRACC) in PBMCs from three groups (Figure 2A). In AMI group mRNA expressions of the genes encoding CD94, NKG2A, CD158, CD161, CD337, CD314 and CD319 were significantly lower than in SAP patients and the controls ($p<0.05$). There was no statistical difference in NK cell biomarker expressions between SAP and the controls. Density curves for the NK cell proportion in PBMCs from three groups were delineated (Figure 2B). The two density curves of cell proportion from AMI and SAP patients in PBMCs were substantially left shift when compared with the controls. The number of NK cells was significantly decreased in both AMI and SAP patients ($p<0.01$). However, there was no significant difference between AMI and SAP patients in the quantity of NK cells.

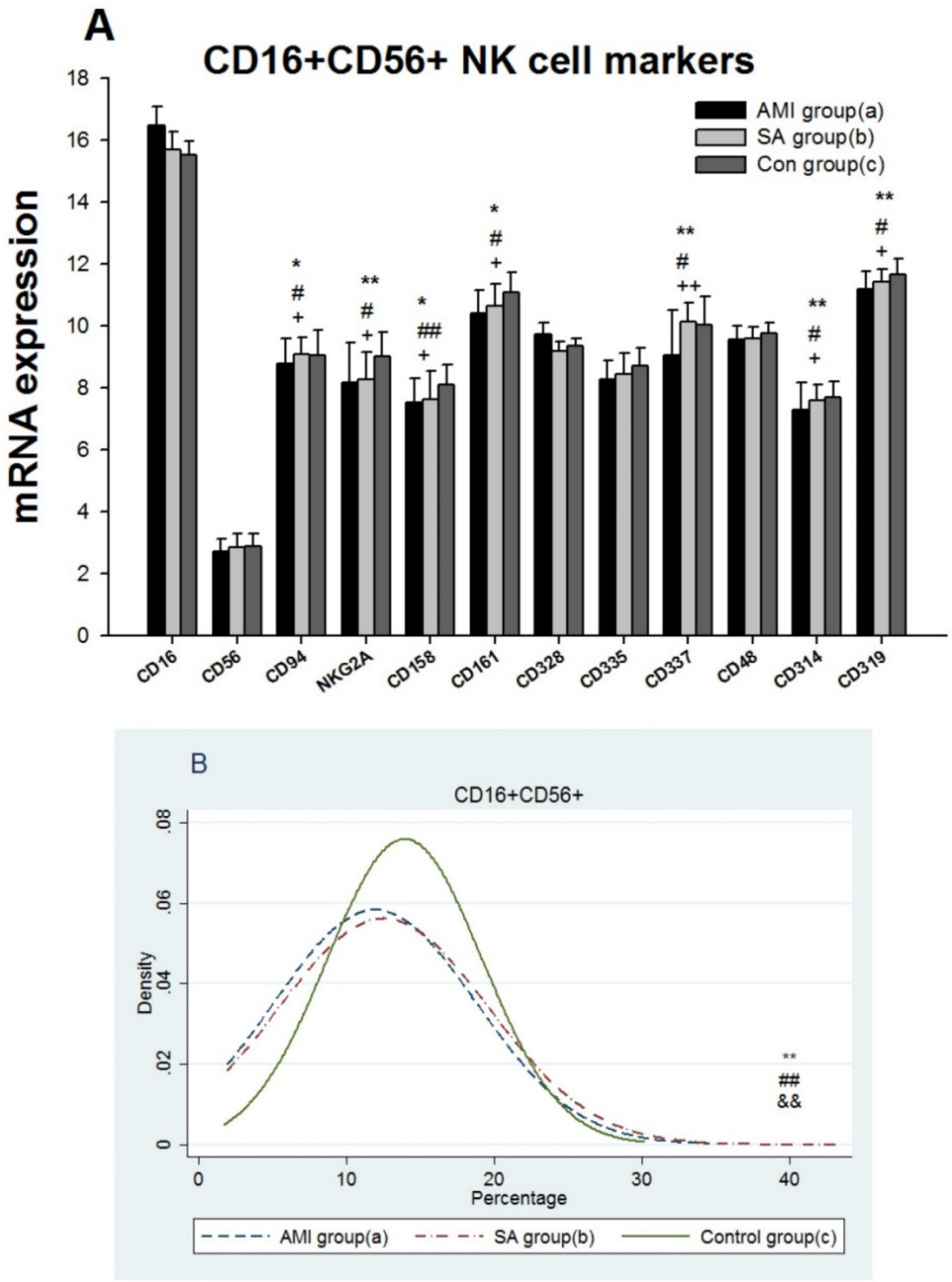


Figure 2. From three groups in PBMCs, (A) mRNA expression of intracellular and extracellular markers of CD16+CD56+cells. (B) The comparison of CD16+CD56+ cells counting. Three groups: *, P<0.05; **, P<0.01. AMI vs. Con: #, P<0.05; ##, P<0.01. AMI vs. SAP: +, P<0.05; ++, P<0.01. SAP vs. Con &: P<0.05; &&: P<0.01.

Gene expression, subsets counting and related cytokines of T cells

Expressions of 8 genes related to T cell receptor (TCR) antigen recognition, 16 genes associated with CD4+T cells and 15 genes with CD8+ T cells were detected among three groups (Figure 3A, 3C, 3E). 16 genes in AMI patients encoding TCRA, TCRB, TCRG, TCRZ, CD3D, CD3E, CD3G, CD195(CCR5), IL-10, GATA3, CD278(ICOS), CD8A, CD8B, CD28, GZMM and CASP10 were significantly down-regulated when compared with the SAP patients and controls respectively($p<0.05$). TCRIM, CD294 (CRTH2) and GZMK expressions in AMI group were significantly lower than those in SAP group ($p<0.05$). Comparing with controls, gene expressions of CD4, IL4 and TNFA in AMI group were significantly down-regulated ($p<0.05$), while IL-2 and CD366 (Tim-3) mRNA expressions were up-regulated ($p<0.05$). Between SAP and control group, there was no statistical difference in TCR, CD4+ and CD8+T cell markers related mRNA expression.

Results from the proportions of cytological T lymphocyte subsets in PBMCs among three groups showed the levels of CD3+ and CD8+T cells in AMI and SAP group decreased significantly ($p< 0.05$), while CD4+T cells increased ($p<0.01$) when compared with control group (Figure 3B, 3D, 3F, Table 2). The cytokine IL-2, IL-4, IL-6 and IFN were significantly increased in AMI and SAP patients when compared with the controls ($p<0.01$). However, there was no significant difference between AMI and SAP patients in IL-2, IFN, CD3+ T cell and CD8+T cell quantity (Table 2). The counting of CD4+ T cell, IL-4 and IL-6 were higher in AMI patients than in SAP patients ($p<0.01$).

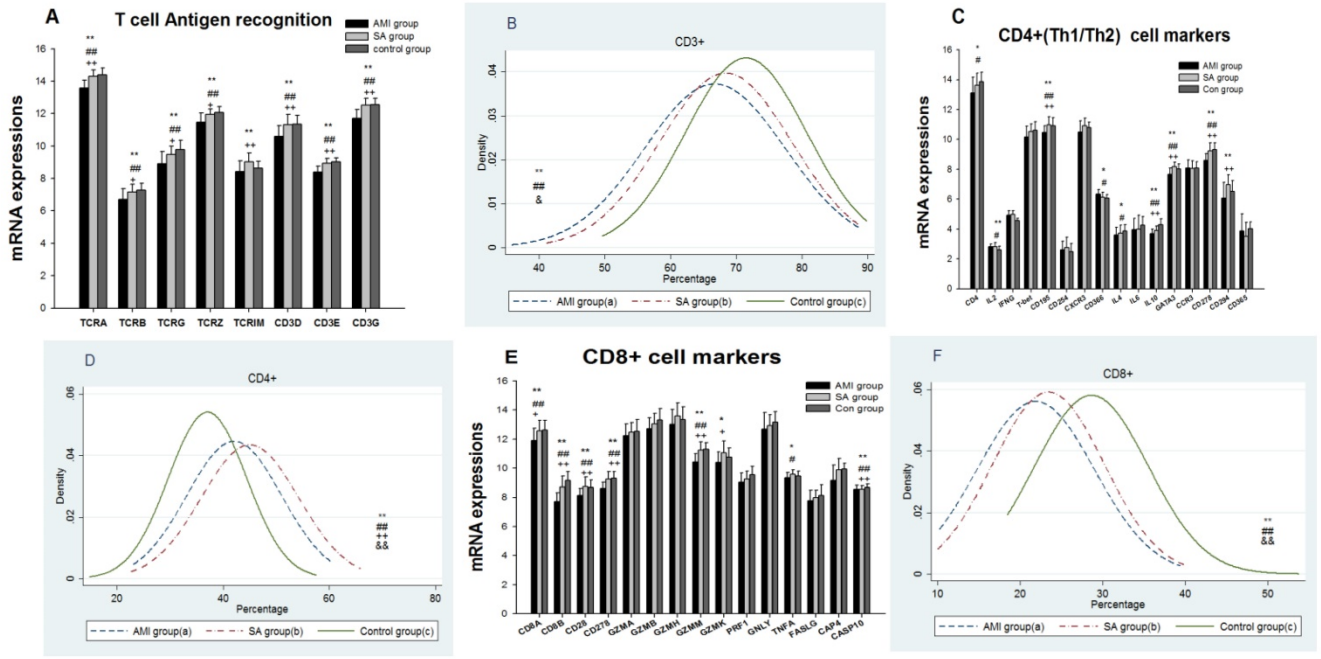


Figure 3. From three groups in PBMCs, (A) Expression of genes related to T cell antigen recognition. (B) CD3+ counting. (C) Expression of genes related to CD4+. (D) CD4+ counting. (E) Genes related to CD8+. (F) CD8+ counting. Three groups: *, P<0.05; **, P<0.01. AMI vs. Con: #, P<0.05; ##, P<0.01. AMI vs. SAP: +, P<0.05; ++, P<0.01. SAP vs. Con &: P<0.05; &&: P<0.01.

Table 2. Values of T cell immunity among three groups ($\bar{x}$ ± s.d.).

Index	AMI (a) (N=210)	SAP (b) (N=210)	Con(c) (N=250)	P (all)	P (a/c)	P (b/c)	P (a/b)
CD3+ (%)	66.7±10.7	68.4±10.0	71.5±9.2	0.00	0.00	0.002	0.275
CD4+ (%)	42.1±8.9	45.1.±9.2	37.0±9.1	0.00	0.00	0.00	0.003
CD8+ (%)	21.8±7.1	23.3±6.7	28.6±6.9	0.00	0.00	0.00	0.068
IL-2 (pg/ml)	34.2±18.5	33.5±14.5	9.9±2.3	0.00	0.00	0.00	0.96
IL-4(pg/ml)	35.7±15.7	28.22±10.9	4.8±2.3	0.00	0.00	0.00	0.00
IL-6 (pg/ml)	29.9±16.2	24.6±14.4	3.0±1.4	0.00	0.00	0.00	0.001
IFN (pg/ml)	40.8±21.4	33.0±22.1	16.7±6.3	0.00	0.00	0.00	0.72

Gene expression, counting and serum immunoglobulin level of B cells

The results showed that expressions of 15 genes related to B cell biomarkers in patients with AMI, SAP and the controls (Figure 4A), including CD5, CD19, CD20, CD21 (CR1), CD22, CD23, CD40, CD79a, CD79b, CD80(B7-1), CD86(B7-2), CD138, CD154(IgM), CD268(BAFFR) and CD279(PD-1). In PBMCs from three groups, expressions of 8 genes encoding CD5, CD19, CD20, CD22, CD40, CD79b, CD268 and CD279 in AMI group were significantly lower than those from SAP and control group ($p<0.05$). Compared with controls, gene expressions of CD21, CD23 and CD79a were significantly down-regulated in AMI patients ($p<0.05$). Between the SAP and control group, there were no significant differences in B cell marker expressions. When compared with controls, B cell counting, IgG and IgM in PBMCs were significantly down-regulated ($p<0.01$), while IgA was significantly increased in both AMI and SAP group (Figure 4B, Table 3) ($p<0.05$).

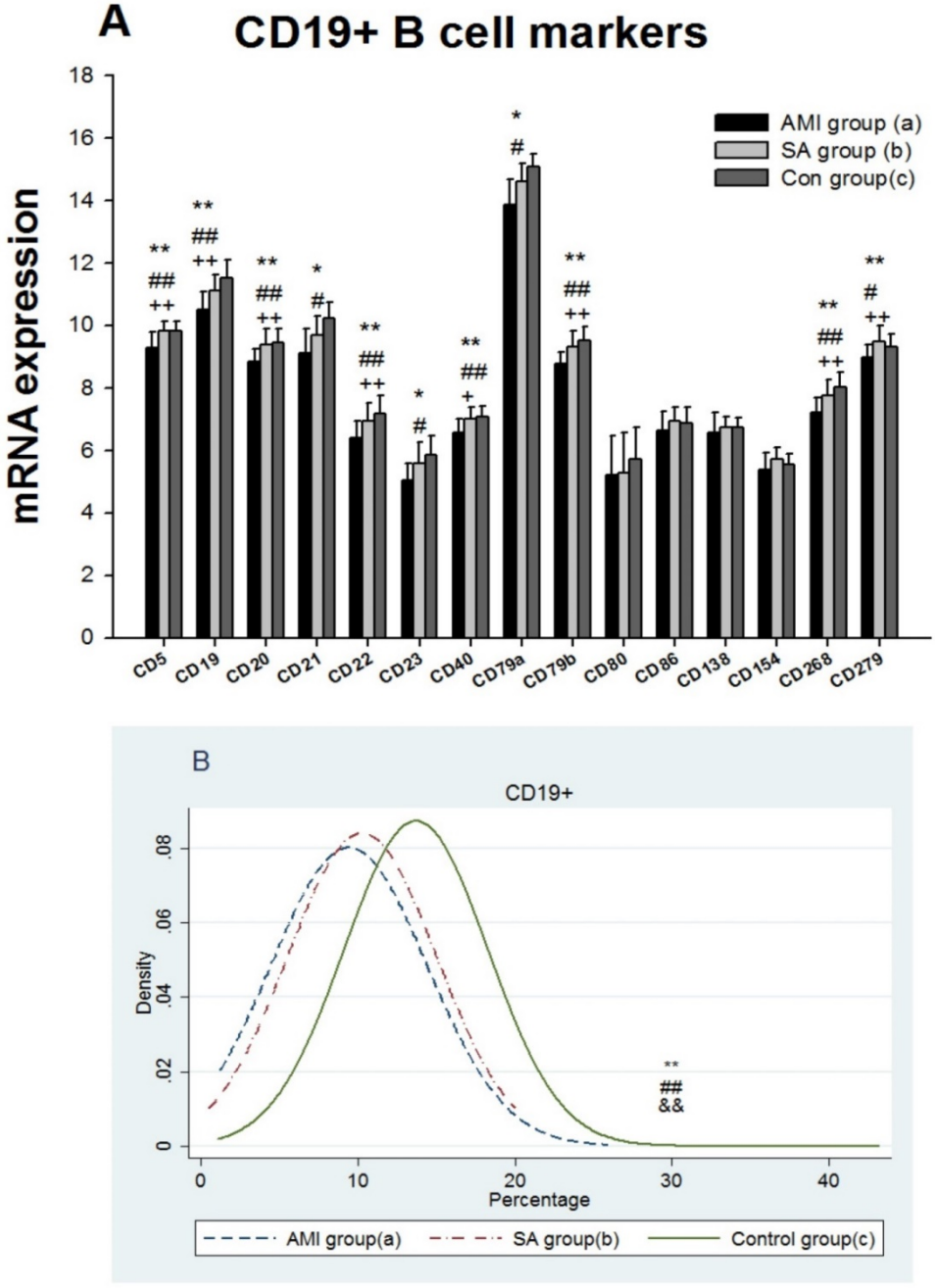


Figure 4. From three groups in PBMCs, (A) mRNA expression of intracellular and extracellular markers of CD19+ cell. (B) The comparison of CD19+ cells counting. Three groups: *, P<0.05; **, P<0.01. AMI vs. Con: #, P<0.05; ##, P<0.01. AMI vs. SAP: +, P<0.05; ++, P<0.01. SAP vs. Con &: P<0.05; &&: P<0.01.

Table 3. Values of B cell immunity among three groups ($\bar{x}$ ± s.d.).

Index	AMI(a) (N=210)	SAP (b) (N=210)	Con(c) (N=250)	P (all)	P (a/c)	P (b/c)	P (a/b)
$CD19^+$(%)	9.4±5.0	10.3±4.7	13.7±4.6	0.00	0.00	0.00	0.212
IgA (g/L)	2.3±1.0	2.2±0.8	1.9±0.7	0.00	0.001	0.014	0.487
IgM(g/L)	0.8±0.42	0.8±0.35	1.2±0.41	0.00	0.00	0.00	0.572
IgG (g/L)	10.8±2.6	11.2±2.2	12.0±2.3	0.00	0.00	0.001	0.242

Discussion

In our current study, the significantly up-regulated mRNAs expressions of early complement components, C1qá, C1qâ, C1qɤ, C1r and C5a demonstrated that the classical pathways were activated in AMI patients. The initiation of classical pathway eventually results in the terminal access to form C5b-9 complex, which makes a transmembrane pore in the target cells' membrane to lysis [17]. C5b initiates the formation of MAC, which consists of C5b, C6, C7, C8, and multiple molecules of C9. In our study the significantly lowest levels of C7, C8â and C9 mRNAs in AMI patients suggested the obstacle of MAC formation. In AMI and SAP patients, the serum levels of C3, C4 and CH50, which reflected the activities of C1-C9 via classic pathway, were all elevated. Gene and cytology levels of the complement in both AMI and SAP patients were activated and the results were consistent with previous studies [18, 19, 20]. Though the complement was activated in AMI and SAP stages, based on the genomics results of complement cascade reaction imbalance, cytolytic effect of the complement only decreased in AMI patients.

NK cells express an array of inhibitory and activating receptors. The inhibitory receptors are responsible for self-tolerance while activating receptors mediate the NK cell cytotoxicity (NKCC) [21, 22]. KIRs are the most important NK cell receptors, including CD94, NKG2A, CD158 and CD314, which recognize classical MHC class I [23]. In present study, the gene expressions of CD94, NKG2A and CD158 were significantly lower than those in SAP patients and the controls, suggesting the impaired ability to protect normal cells in AMI patients. Receptors CD335 (NKp46), CD337 (NKp30), CD48 (2B4), CD314(NKG2D) and CD319 (CRACC) are most central activating receptors and play an important role in targeting NK cell responses toward abnormal cells and eventually the cell lysis [24, 25, 26, 27]. In our current study, gene expressions of activating receptors, CD337, CD314 and CD319 in AMI patients were significantly decreased in comparison with SAP patients and controls respectively, which showed the transduction of activating signal was inhibited in patients with AMI. The cytotoxic ability of NK cells was decreased afterwards. There was no significant difference in mRNA expression between the SAP patients and controls in inhibitory and activating receptors, indicating the NK receptors in SAP patients was in a nearly inactive state. Previous studies found the reduced proportions of NK cells in peripheral blood of CAD, but the reason was still controversial [28, 29, 30, 31]. The similar loss of NK cell numbers in both AMI and SAP patients were also observed in our study (Figure 2B). Together with the notably decreased expression of NK cell biomarkers in AMI patients, different levels of reduced immunity in NK cells were demonstrated in AMI and SAP stages. In AMI patients both numbers and receptor activity were decreased, while only a deficit of quantity was found in SAP patients.

TCR is a molecule found on the surface of T lymphocytes that is responsible for recognizing antigens. The first signal for T cell activation is provided through the TCR-CD3 [32]. In present study, gene expressions of TCRA, TCRB, TCRG, TCRZ, CD3D, CD3E and CD3G were significantly lower in AMI group than those in SAP and control group (Figure 4A), indicating the decreased ability of TCR antigen recognition. In addition, the loss of CD3+ T cells in PBMCs was found in both AMI and SAP patients (Figure 4B), suggesting the dysfunction of CD3+T cells in CAD, especially in AMI stage.

Naive CD4+ T cells differentiate into T helper type 1 (Th1) and T helper type 2 (Th2). Th1 cells achieve cellular immunity mainly by secreting IL2, IL12 and IFN-ã. T-bet is a Th1 transcription factor for regulating Th1 development [33]. CD195 (CCR5) and CD182 (CXCR3) are specific Th1 lymphocytes chemokine receptors [34]. Th2 cells produce IL4, IL6 and IL10 to activate B lymphocytes and generate antibodies. GATA3 is the Th2 specific transcription factor, and CCR3 together with CD294 (CRTH2) are chemokine receptors of Th2 cells [35, 36, 37]. CD366 (Tim-3) is a Th1-specific cell surface protein while CD365 (Tim-1) is Th2-specific [38, 39]. The high mRNA expressions of Th1 biomarkers (IL2 and CD366) and low RNA expressions of Th2 biomarkers (IL4, IL-10, CD278 and CD294) in AMI patients suggested a shift towards Th1 dominance. The significant increase of CD4+T cells, IL-4 and IL-6 in

AMI than in SAP patients showed the differential degrees of CD4+ T cell mediated cellular immunity dysfunction in AMI and SAP patients.

CD8+ T cells kill virus-infected cells and tumor cells and play a critical role in immune protection [40]. CD8+T cell is firstly activated by TCR and CD8 binding and then co-stimulatory molecules.CD8+ T cells make the fatal attack through the perforin-granzyme, Fas-Fas ligand (FasL), and TNF-a pathways [41,42]. The presence of CD8+ T cells in atherosclerotic lesions is widely demonstrated but studies investigating their role in atherogenesis have yielded contradictory results [43, 44]. In the present study, all 15 genes related to killing ability of CD8+ T cells in AMI patients were down-regulated; especially CD8A, CD8B, CD28, CD278, GZMK, GZMM, PRF1 and CASP10 were significantly down-regulated when compared with SAP and/or controls. Together with significant loss of CD8+T cells in AMI and SAP patient in PBMCs indicated the decreased cytotoxic ability of CD8+ T cells in CAD patients, particularly in the stages of AMI patients.

We detected all 15 genes related with intracellular and extracellular markers of CD19+B cells [45] (Figure 4A). B cell receptor (BCR) was composed of membrane immunoglobulin (Ig) which recognizes the antigens while Igá (CD79A) and Igâ (CD79B) transmit the activation signals [46]. CD19 and CD21 are B cell co-receptors and enhance the BCR signal transduction [47]. The B cell specific Src-family kinase CD5, specifically binding B cell surface Ig, is dispensable for B cell activation [48]. CD268 is the principal receptor required for BAFF-mediated B cell activation [49]. CD279 encodes a cell surface membrane protein of Ig superfamily and plays a role in their differentiation [50]. In AMI patients, gene expressions of CD5, CD19, CD20, CD21, CD22, CD23, CD40, CD79A, CD79B, CD268 and CD279 were significantly lower than those in SAP and /or control group, which showed B cell activation were blocked in AMI patients. There was no significant gene expression difference in B cell activation between the SAP patients and the control group. The detection of B cell quantity, IgM and IgG levels in PBMCs were decreased in both AMI and SAP patients. In sum, in AMI patients, gene expressions and numbers of B cells were reduced, demonstrating the deeply weakened humoral immunity in AMI group.

In the present study, in AMI patients the mRNA expression of immune system was consistent with the cytological level and the decline of both parameters demonstrated the collapse of immune function in AMI group. In SAP patients, the immunity related gene expression was different from cytological level. The CH50, C3 and C4 were increased and the number of NK cells, CD3+, CD8+ T cells and CD19+ B cells were decreased, while the gene expression of immune system was in a nearly inactive status. In the current study, we can conclude that the attack of AMI and SAP was associated with different levels of immune dysfunction. AMI occurred in the stage of immune collapse while SAP occurred in progressively reduced level of immunity but still within the boundary of compensation. The quantity of immune cells in peripheral blood may reflect the current state of immune function and the gene expression of immune system stands for the compensatory capacity of the immune system.

In AMI patients, the suppressed innate and adaptive immune system, especially the cytotoxic ability, failed to remove the exogenous pathogens. Various exogenous microorganism infections are supposed as risk factors of AMI [8, 9] and infection seems to be linked to plaque rupture [4, 5, 6, 7].

Conclusions

The pathogenesis of AMI might be related with infections of pathogens under the depletion of immune system. That is the reason why single vaccine is ineffective on AMI prevention. To improve the immunity of CAD patients may be considered as a potential target for medical intervention and prevention of AMI.

Acknowledgements

The study was supported by Shanghai Traditional Chinese Medicine 3-year Development Program (2014-2016); National Natural Science Foundation (81570359) and Shanghai municipal health and Family Planning Commission project (20144Y0046).

Competing Interests

The authors have declared that no competing interest exists.

References

[1] Libby P, Ridker PM, Hansson GK. Progress and challenges in translating the biology of atherosclerosis. Nature. 2011; 473: 317–325.
[2] Libby P, Theroux P. Pathophysiology of coronary artery disease. Circulation. 2005; 111: 3481-3488.
[3] Shah PK. Pathophysiology of coronary thrombosis: role of plaque rupture and plaque erosion. Prog Cardiovasc Dis. 2002; 44:357-368.
[4] Levi M, van der Poll T, Schultz M. New insights into pathways that determine the link between infection and thrombosis. Neth J Med. 2012; 70:114-120.
[5] Chatzidimitriou D, Kirmizis D, Gavriilaki E, et al. Atherosclerosis and infection: is the jury still not in? Future Microbiol. 2012; 7:1217-1230.
[6] Levi M, van der Poll T, Schultz M. Infection and inflammation as risk factors for thrombosis and atherosclerosis. Semin Thromb Hemost. 2012; 38:506-514.
[7] Khan S, Rahman HN, Okamoto T, et al. Promotion of atherosclerosis by Helicobacter cinaedi infection that involves macrophage-driven proinflammatory responses. Sci Rep. 2014; 4: 4680.
[8] Mostafa A, Mohamed MK, Saeed M, et al. Hepatitis C infection and clearance: impact on atherosclerosis and cardiometabolic risk factors. Gut. 2010; 59:1135-1140.

[9] Yan WW, Zhang KS, Duan QL, et al. Significantly reduced function of T cells in patients with acute arterial thrombosis. Journal of Geriatric Cardiology. 2015; 12: 287-293.
[10] Yan WW, Wang LM, Jiang JF, et al. Differential expression of T cell-related genes in AMI and SA stages of coronary artery disease. Int J ClinExp Med. 2015; 8:10875-10884.
[11] Matusik P, Guzik B, Weber C, et al. Do we know enough about the immune pathogenesis of acute coronary syndromes to improve clinical practice? Thromb Haemost. 2012; 108:443-456.
[12] Arbab-Zadeh A, Nakano M, Virmani R, et al. Acute coronary events. Circulation. 2012; 125: 1147–1156.
[13] Dobbin K, Simon R. Sample size determination in microarray experiments for class comparison and prognostic classification. Biostatistics. 2005; 6:27-38.
[14] Thygesen K, Alpert JS, Jaffe AS, et al. Third universal definition of myocardial infarction. J Am Coll Cardiol. 2012; 60: 1581-1598.
[15] Wiltgen M, Tilz GP. DNA microarray analysis: principles and clinical impact. Hematology. 2007; 12:271-287.
[16] Cooper MA, Colonna M, Yokoyama WM. Hidden talents of natural killers: NK cells in innate and adaptive immunity. EMBO Rep. 2009; 10:1103-1110.
[17] Lappegård KT, Garred P, Jonasson L, et al. A vital role for complement in heart disease. Mol Immunol. 2014; 61:126-134.
[18] Iltumur K, Karabulut A, Toprak G, et al. Complement activation in acute coronary syndromes. APMIS. 2005; 113: 167-174.
[19] Cusack MR, Marber MS, Lambiase PD, et al. Systemic inflammation in unstable angina is the result of myocardial necrosis. J Am Coll Cardiol. 2002; 39:1917–1923
[20] Leinoe E, Pachai A, Brandslund I. Complement activation reaches maximum during equilibrium between antigen and antibody in an in vitro model for thrombolysis with streptokinase. APMIS. 2000; 10:685–688.
[21] Backström E, Kristensson K, Ljunggren HG. Activation of natural killer cells: underlying molecular mechanisms revealed. Scand J Immunol. 2004; 60: 14-22.
[22] Middleton D, Curran M, Maxwell L. Natural killer cells and their receptors. Transpl Immunol. 2002; 10: 147-64.
[23] Campbell KS, Purdy AK. Structure/function of human killer cell immunoglobulin-like receptors: lessons from polymorphisms, evolution, crystal structures and mutations. Immunology. 2011; 132:315-325.
[24] Raulet DH, Gasser S, Gowen BG, et al. Regulation of ligands for the NKG2D activating receptor. Annu Rev Immunol. 2013; 31:413-441.
[25] Zafirova B, Wensveen FM, Gulin M, et al. Regulation of immune cell function and differentiation by the NKG2D receptor. Cell Mol Life Sci. 2011; 68:3519-3529.
[26] Koch J, Steinle A, Watzl C, et al. Activating natural cytotoxicity receptors of natural killer cells in cancer and infection. Trends Immunol. 2013; 34:182-191.
[27] Claus M, Meinke S, Bhat R, et al. Regulation of NK cell activity by 2B4, NTB-A and CRACC. Front Biosci. 2008; 13: 956-965.
[28] Whitman SC, Rateri DL, Szilvassy SJ, et al. Depletion of natural killer cell function decreases atherosclerosis in low-density lipoprotein receptor null mice. Arterioscler Thromb Vasc Biol. 2004; 24:1049-1054
[29] Li W, Lidebjer C, Yuan XM, et al. NK cell apoptosis in coronary artery disease: relation to oxidative stress. Atherosclerosis. 2008; 199:65-72.
[30] Jonasson L, Backteman K, Ernerudh J. Loss of natural killer cell activity in patients with coronary artery disease. Atherosclerosis. 2005; 183:316-321.
[31] Backteman K, Ernerudh J, Jonasson L. Natural killer (NK) cell deficit in coronary artery disease: no aberrations in phenotype but sustained reduction of NK cells is associated with low-grade inflammation. Clin Exp Immunol. 2014; 175:104-112.
[32] Zehn D, King C, Bevan MJ, et al. TCR signaling requirements for activating T cells and for generating memory. Cell Mol Life Sci. 2012; 69: 1565-1575.
[33] Vanaki E, Ataei M, Sanati MH, et al. Expression patterns of Th1/Th2 transcription factors in patients with guttate psoriasis. Acta Microbiol Immunol Hung. 2013; 60: 163-174.
[34] Gao P, Zhou XY, Yashiro-Ohtani Y, et al. The unique target specificity of a nonpeptide chemokine receptor antagonist: selective blockade of two Th1 chemokine receptors CCR5 and CXCR3. J Leukoc Biol. 2003; 73: 273-280.
[35] Wan YY. GATA3: a master of many trades in immune regulation. Trends Immunol. 2014; 35: 233-242.
[36] Sallusto F, Mackay CR, Lanzavecchia A. Selective expression of the eotaxin receptor CCR3 by human T helper 2 cells. Science. 1997; 277: 2005-2007.
[37] Nagata K, Hirai H. The second PGD< sub> 2</sub> receptor CRTH2: structure, properties, and functions in leukocytes. Prostaglandins Leukot Essent Fatty Acids. 2003; 69: 169-177.
[38] Hastings WD, Anderson DE, Kassam N, et al. TIM-3 is expressed on activated human CD4+ T cells and regulates Th1 and Th17 cytokines. Eur J Immunol. 2009; 39:2492-2501.
[39] Curtiss ML, Gorman JV, Businga TR, et al. Tim-1 regulates Th2 responses in an airway hypersensitivity model. Eur J Immunol. 2012; 42:651-661.
[40] Gadhamsetty S, Marée AFM, Beltman JB, et al. A General Functional Response of Cytotoxic T lymphocyte –Mediated Killing of Target Cells. Biophys J. 2014; 106: 1780-1791.
[41] Keefe D, Shi L, Feske S, et al. Perforin triggers a plasma membrane-repair response that facilitates CTL induction of apoptosis. Immunity. 2005; 23: 249-262.
[42] Berke G. The CTL's kiss of death. Cell. 1995; 81: 9-12.
[43] Kyaw T, Winship A, Tay C, et al. Cytotoxic and proinflammatory CD8+ T lymphocytes promote development of vulnerable atherosclerotic plaques in apoE-deficient mice. Circulation. 2013; 127:1028-1039
[44] Zhou J, Dimayuga PC, Zhao X, et al. CD8 (+) CD25 (+) T cells reduce atherosclerosis in apoE (-/-) mice. Biochem Biophys Res Commun. 2014; 443:864-870.
[45] Pike KA, Ratcliffe MJ. Cell surface immunoglobulin receptors in B cell development. Semin Immunol. 2002; 14:351-358.
[46] Kurosaki T. Regulation of BCR signaling. Mol Immunol. 2011; 48:1287-1291.
[47] Barrington RA, Schneider TJ, Pitcher LA, et al. Uncoupling CD21 and CD19 of the B-cell coreceptor. Proc Natl Acad Sci USA. 2009; 106:14490-14495.
[48] Pospisil R, Silverman GJ, Marti GE, et al. CD5 is A potential selecting ligand for B-cell surface immunoglobulin: a possible role in maintenance and selective expansion of normal and malignant B cells. Leuk Lymphoma. 2000; 36:353-365.
[49] Bergmann H, Yabas M, Short A, et al. B cell survival, surface BCR and BAFFR expression, CD74 metabolism, and CD8- dendritic cells require the intramembrane endopeptidase SPPL2A. J Exp Med. 2013; 210: 31-40.
[50] Thibult ML, Mamessier E, Gertner-Dardenne J, et al. PD-1 is a novel regulator of human B-cell activation. Int Immunol. 2013; 25:129-137.

Preordering using a Target-Language Parser via Cross-Language Syntactic Projection for Statistical Machine Translation

ISAO GOTO, National Institute of Information and Communications Technology, NHK, and Kyoto University
MASAO UTIYAMA and EIICHIRO SUMITA, National Institute of Information and Communications Technology
SADAO KUROHASHI, Kyoto University

When translating between languages with widely different word orders, word reordering can present a major challenge. Although some word reordering methods do not employ source-language syntactic structures, such structures are inherently useful for word reordering. However, high-quality syntactic parsers are not available for many languages. We propose a preordering method using a target-language syntactic parser to process source-language syntactic structures without a source-language syntactic parser. To train our preordering model based on ITG, we produced syntactic constituent structures for source-language training sentences by (1) parsing target-language training sentences, (2) projecting constituent structures of the target-language sentences to the corresponding source-language sentences, (3) selecting parallel sentences with highly synchronized parallel structures, (4) producing probabilistic models for parsing using the projected partial structures and the Pitman-Yor process, and (5) parsing to produce full binary syntactic structures maximally synchronized with the corresponding target-language syntactic structures, using the constraints of the projected partial structures and the probabilistic models. Our ITG-based preordering model is trained using the produced binary syntactic structures and word alignments. The proposed method facilitates the learning of ITG by producing highly synchronized parallel syntactic structures based on cross-language syntactic projection and sentence selection. The preordering model jointly parses input sentences and identifies their reordered structures. Experiments with Japanese–English and Chinese–English patent translation indicate that our method outperforms existing methods, including string-to-tree syntax-based SMT, a preordering method that does not require a parser, and a preordering method that uses a source-language dependency parser.

Categories and Subject Descriptors: I.2.7 [**Artificial Intelligence**]: Natural Language Processing—*Machine translation*

General Terms: Theory, Algorithms, Design, Experimentation

Additional Key Words and Phrases: Preordering, syntactic projection, constituent structure, inversion transduction grammar

ACM Reference Format:
Goto, I., Utiyama, M., Sumita, E., and Kurohashi, S. 2015. Preordering using a target-language parser via cross-language syntactic projection for statistical machine translation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 14, 3, Article 13 (June 2015), 23 pages.
DOI: http://dx.doi.org/10.1145/2699925

Authors' addresses: I. Goto (corresponding author), NHK Science & Technology Research Laboratories, 1-10-11 Kinuta, Setagaya-ku, Tokyo 157-8510, Japan; email: goto.i-es@nhk.or.jp; M. Utiyama and E. Sumita, Multilingual Translation Laboratory, National Institute of Information and Communications Technology, 3-5 Hikaridai, Keihanna Science City, Kyoto 619-0289, Japan; emails: {mutiyama, eiichiro.sumita}@nict.go.jp; S. Kurohashi, Department of Intelligence Science and Technology, Graduate School of Informatics, Kyoto University, Yoshida-honmachi, Sakyo-ku, Kyoto 606-8501, Japan; email: kuro@i.kyoto-u.ac.jp.

2375-4699/2015/06-ART13 $15.00
DOI: http://dx.doi.org/10.1145/2699925

1. INTRODUCTION

Estimating the appropriate word order for a target language is one of the most difficult problems in statistical machine translation (SMT). This is particularly true when translating between languages with widely different word orders, such as Japanese and English. To address this, a large body of research has been conducted on word reordering, for instance: lexicalized reordering model [Tillman 2004] for phrase-based SMT, hierarchical phrase-based SMT [Chiang 2007], syntax-based SMT [Yamada and Knight 2001], preordering [Xia and McCord 2004], and postordering [Sudoh et al. 2011b].

The preordering framework is useful for word reordering because it can use source-language syntactic structures in a simple way. Specifically, a preordering method using source-language syntactic structures for English-to-Japanese translation has been confirmed to be highly effective [Goto et al. 2011; Sudoh et al. 2011a]. Existing preordering methods that use source-language syntactic structures require a source-language syntactic parser. Unfortunately, high-quality syntactic parsers are not available for many languages.

Preordering methods that do not require a parser are useful in cases where no source-language syntactic parser is available [DeNero and Uszkoreit 2011; Neubig et al. 2012]. Such methods produce preordering rules using word alignments. However, these preordering rules do not use syntactic structures, which are an essential factor in deciding word order. Therefore, the use of syntactic structures is a major challenge for preordering methods that do not require a source-language syntactic parser.

In this article, we propose a novel preordering approach that uses syntactic structures by employing a target-language syntactic parser without requiring a source-language parser. A high-quality target-language constituency parser will be useful for preordering. Source-language syntactic structures and corresponding target-language syntactic structures are expected to be similar in a parallel corpus [Hwa et al. 2005]. The proposed method relies on this expectation. We project target-language syntactic constituent structures in a parallel corpus onto their corresponding source-language sentences through word alignments, which produces partial syntactic structures where the words are from the source language but the phrase labels are from the target-language syntax. We then select parallel sentences with highly synchronized parallel syntactic structures based on the projection. We construct a probabilistic context-free grammar (CFG) model and a probabilistic model for unsupervised part-of-speech (POS) tagging using the partial syntactic structures of the selected parallel sentences and the Pitman-Yor process [Pitman and Yor 1997]. We then parse the source-language training sentences to produce full binary syntactic tree structures using the produced probabilistic models with the projected partial syntactic structure constraints. A preordering model based on inversion transduction grammar (ITG) [Wu 1997] is learned using the full binary syntactic constituent structures of the source-language sentences and word alignments. Input sentences are parsed using the ITG-based preordering model, then their syntactic structures and reordered structures are identified jointly.

Our main contributions are (i) a new effective framework for preordering using a target-language syntactic parser that does not require a source-language syntactic parser, (ii) methods that facilitate the learning of ITG by producing highly synchronized parallel syntactic structures based on cross-language syntactic projection and sentence selection, (iii) a simple method for producing full binary syntactic constituent structures of source-language sentences from the constituent structures of the corresponding target-language sentences using the Pitman-Yor process, and (iv) an

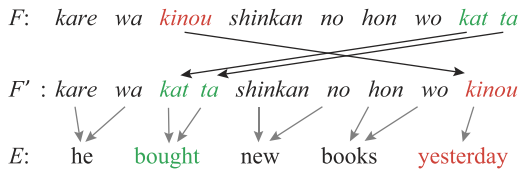


Fig. 1. Example of preordering for Japanese–English translation.

empirical confirmation of the efficacy of Japanese–English and Chinese–English patent translation.

There is a need for translation in situations where all the following are true: (1) a high-quality target-language parser is available, (2) a high-quality source-language parser is not available, and (3) the source-language word order and the target-language word order are largely different, such as in subject-object-verb (SOV) and subject-verb-object (SVO) languages. We propose a method that can be applied in such situations. In our experiments on Japanese–English and Chinese–English translation using the patent data from the NTCIR-9 and NTCIR-10 Patent Machine Translation Tasks [Goto et al. 2011, 2013a], we were able to significantly improve translation quality, as measured by both RIBES [Isozaki et al. 2010] and BLEU [Papineni et al. 2002]. Our method is superior to phrase-based SMT, hierarchical phrase-based SMT, string-to-tree syntax-based SMT, an existing preordering method without a parser, and an existing preordering method that uses a source-language dependency parser.

The rest of this article is organized as follows: Section 2 describes the preordering framework and previous work; Section 3 provides an overview of the proposed method; Section 4 explains the training method; Section 5 describes the preordering method; Section 6 reports and discusses the experimental results; and Section 7 concludes the article.

2. PREORDERING FOR SMT

Machine translation is defined as the process in which a source-language sentence F is transformed into a target-language sentence E. During this process, word reordering is often necessary. More specifically, long-distance word reordering is necessary when translating between languages with widely different word orders.

The syntactic structure of F is useful for long-distance word reordering. Preordering is an SMT method in which the syntactic structure of F can be handled in a simple way. This is the approach that we take in this article. In the preordering approach, translation is performed as a two-step process, as shown in Figure 1. In the first part of the process, F is reordered to F', which is a source-language word sequence with almost the same word order as the target language. In the second part of the process, F' is translated into E using an SMT method such as phrase-based SMT, which can produce accurate translations when only local reordering is required.

The preordering framework has been widely studied. In most preordering research, the reordering of words is conducted using reordering rules and the syntactic structure of F that is obtained using a source-language syntactic parser. Reordering rules are produced automatically [Xia and McCord 2004; Li et al. 2007; Habash 2007; Dyer and Resnik 2010; Ge 2010; Genzel 2010; Visweswariah et al. 2010; Wu et al. 2011a, 2011b; Lerner and Petrov 2013] or manually [Collins et al. 2005; Wang et al. 2007; Ramanathan et al. 2008; Badr et al. 2009; Xu et al. 2009; Isozaki et al. 2012; Hoshino et al. 2013].

However, if a source-language syntactic parser is not available, then these methods cannot be applied. In such cases, preordering methods that do not require a parser

Table I. Comparison of Preordering Methods based on the Necessity of Syntactic Parsers for Source and Target Languages

Preordering methods	Parser	
	Source	Target
Most existing methods	✓	
[Neubig et al. 2012]		
Proposed method		✓

are useful [Tromble and Eisner 2009; Visweswariah et al. 2011; DeNero and Uszkoreit 2011; Neubig et al. 2012; Khapra et al. 2013].

Methods that induce a parser deserve particular mention because they are similar to our approach. DeNero and Uszkoreit [2011] and Neubig et al. [2012] induce a nonsyntactic parser automatically using a parallel corpus with word alignments. The induced nonsyntactic parser is used to produce binary tree structures of input sentences. The input sentences are then preordered based on the binary tree structures and bracketing transduction grammar (BTG) [Wu 1997]. The resulting binary tree structures are nonsyntactic structures. In contrast, our method utilizes syntactic structures for preordering via a target-language syntactic parser.

Compared with the nonsyntactic structures that are produced by a nonsyntactic parser based on BTG [Neubig et al. 2012], syntactic structures are thought to be superior when making decisions about word reordering for the following two reasons.

— In syntactic structures, a subtree span is expected to be consistent with the span of an expression that has cohesive meanings. For example, clauses are thought to be spans with cohesive meanings, and clauses are expressed by subtrees in syntactic structures. In contrast, in nonsyntactic structures produced by BTG, a subtree span is not always consistent with the span of an expression with cohesive meanings.
— Syntactic structures are richer in terms of information than nonsyntactic structures produced by BTG. Syntactic structures have many phrase label types. In contrast, BTG has only one phrase label type.

Therefore, syntactic structures are thought to be useful when performing word reordering for preordering methods.

Table I compares the necessity of syntactic parsers in existing preordering methods for source and target languages with that of the proposed preordering method. In some cases, a high-quality syntactic parser is not available for the source language, but a high-quality syntactic parser is available for the target language, while the source-language word order and the target-language word order are largely different, such as with SOV and SVO languages. Our method is applicable in these cases.

3. OVERVIEW OF THE PROPOSED METHOD

In this section, we provide an overview of our preordering method.

Our preordering method processes syntactic structures using a target-language parser even when a source-language parser is not available. The syntactic structures of source-language sentences and the syntactic structures of the corresponding target-language sentences are expected to be similar in a parallel corpus [Hwa et al. 2005]. We used this expectation to produce syntactic constituent structures of source-language sentences that are similar to the syntactic constituent structures of the corresponding target-language sentences.

To effectively learn ITG or synchronous CFG [Aho and Ullman 1969], it is important that the level of synchrony between parallel syntactic structures is high. This is because ITG or synchronous CFG rules are learned from the synchronized parts of

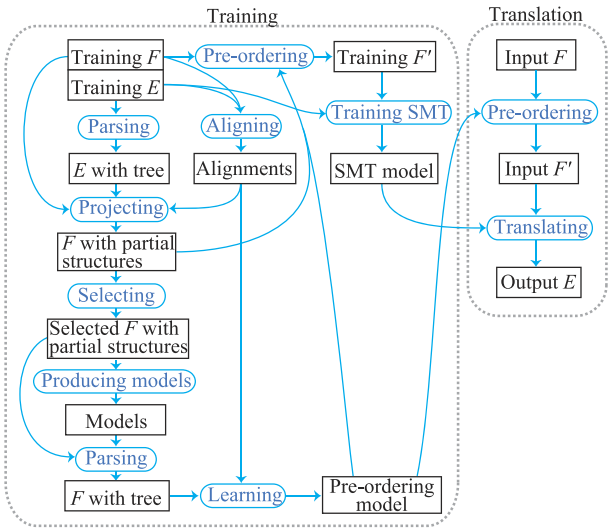


Fig. 2. Overview of our method.

parallel syntactic structures, and such rules cannot be generated from nonsynchronized parts of parallel syntactic structures. That is, it is difficult to effectively learn ITG or synchronous CFG rules from parallel syntactic structures with a low level of synchrony.

Our proposed method facilitates the learning of ITG by producing highly synchronized parallel structures which are then used as training data for ITG, based on the following methods. (i) We produce source-language syntactic structures, which are maximally synchronized with the corresponding target-language syntactic structures, by cross-language syntactic projection. (ii) We select parallel sentences with highly synchronized parallel structures based on cross-language syntactic projection.

Figure 2 shows an overview of our method. Our preordering model is trained via the following steps:

(1) parsing target-language sentences in the parallel training corpus using a syntactic parser to obtain binary[1] tree structures;
(2) projecting the syntactic structures of the training target-language sentences onto the corresponding source-language sentences through word alignments (Section 4.1);
(3) selecting parallel sentences with highly synchronized parallel structures based on the projection (Section 4.2);
(4) producing a probabilistic CFG model and a probabilistic model for unsupervised POS tagging for the source-language using the projected partial syntactic structures (Section 4.3);
(5) parsing the training source-language sentences to produce full binary syntactic tree structures that are highly synchronized with the corresponding target-language syntactic structures. This is conducted using the produced probabilistic models and the projected partial syntactic structures (Section 4.4);
(6) learning the preordering model based on ITG using the full binary syntactic tree structures and word alignments (Section 4.5).

[1]Head binarization is suitable for our method. When trees are not binary trees, our method is applicable if a binarization method is applied.

Input sentences are preordered by jointly parsing and identifying reordering using the ITG-based preordering model.

Our main contribution is an effective new framework for preordering using a target-language parser. Additionally, we propose a new parsing method for source languages that does not require a source-language parser or a source-language POS tagger.

Jiang et al. [2011] developed a method for projecting constituent structures between languages. There are two main differences between our method and theirs. One is the method for estimating CFG rule probabilities. They count the number of CFG rules appearing in tree candidates in each sentence for maximum likelihood estimation of CFG rule probabilities. In this process, they assume a uniform distribution over the projected tree candidates and then calculate the expected counts under this assumption. This looks like a single iteration of the EM algorithm. However, their assumption is incorrect. The expected counts of CFG rules in probable tree candidates should be larger than those of CFG rules in unlikely tree candidates. Our method solves this problem by simply engaging the Pitman-Yor process. The other difference between our method and that of Jiang et al. [2011] is in the requirements. Their method requires source-language POS tags that are produced by a POS tagger. In contrast, our method does not require source-language POS tags.

In Section 4, we describe the training method of our preordering model in detail. In Section 5, we explain the methods for preordering input sentences and the training sentences.

4. TRAINING THE PREORDERING MODEL

In this section, we will explain the five components of the training method for our preordering model, which is employed after the parsing of target-language training sentences is complete.

4.1. Projecting Partial Syntactic Structures

Through word alignments, we project the binary syntactic constituent structures of the target-language sentences in the training parallel corpus onto the corresponding source-language sentences. Partial syntactic structures of the source-language sentences are then obtained. An example of this projection is shown in Figure 3.

The projection is conducted by (1) identifying the span in F corresponding to a subtree span in E through word alignments, and (2) adding the root phrase label of the subtree in E to the span in F. A span in F is the span from the leftmost position to the rightmost position of the source words that are aligned to the target word(s) in the subtree in E. The root phrase label of a projected subtree in E is added to the projected span in F. Note that if any nonaligned words are adjacent to the span in F, then there is a chance that these words should be contained in the span. That is, when there are nonaligned words adjacent to a span in F, there are ambiguities in the span. A projected span that does not contain the adjacent nonaligned words, which is represented by a horizontal solid line in Figure 3, is called a *minimal projected span*. A phrase label is added to a minimal projected span. In Figure 3, a phrase label is not projected to the span covering *kat ta* because the corresponding target expression (bought) is a word instead of a phrase and does not have a phrase label.

To ensure that the projected structures can compose tree structures and consist solely of high-quality structures, we do not project any subtree spans in E when their corresponding spans in F conflict with one other. Here, the conflict is that two subtree spans that do not overlap in E do overlap, except for non-aligned words, when they are projected to F. That is, we only project the subtree spans of E whose corresponding spans of F are also continuous and do not conflict with one other. We choose to project none of the conflicted spans in E.

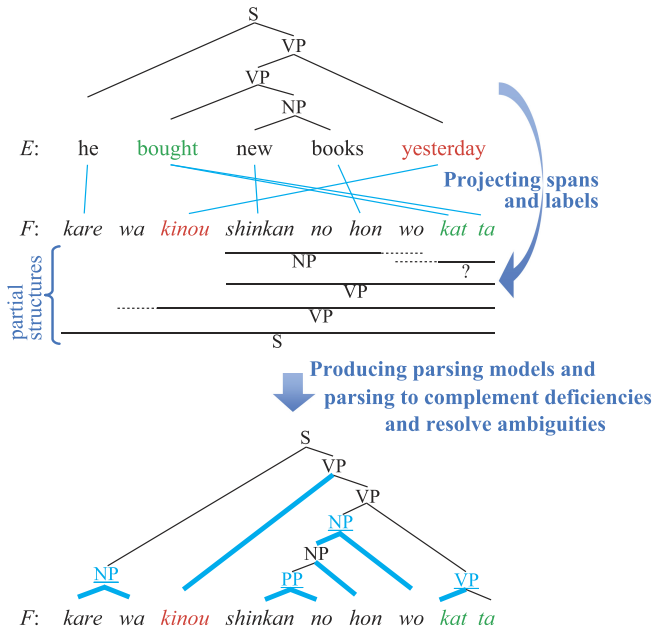


Fig. 3. Example of projecting syntactic structures from E to F and producing a full binary tree structure. The lines between the words in E and the words in F represent word alignments. The horizontal lines represent projected spans and the labels under the horizontal solid lines represent their phrase labels. The dotted lines represent ambiguities in the spans. In the tree structure of F, shown at the bottom part of the figure, the parts that are complemented or resolved ambiguities are represented by underlines for phrase labels and bold lines for structures, which are in blue.

4.2. Selecting Parallel Sentences with Highly Synchronized Structures

We use the projected spans to select parallel sentences with highly synchronized parallel structures. The selected sentences are then used to produce probabilistic models for parsing and full binary tree structures. Let r_1 be the span coverage rate of the projected partial syntactic structures for a source-language sentence. The r_1 for each source-language sentence is calculated by dividing the number of projected spans by the number of words in the sentence minus one.[2] When r_1 is high, the level of synchrony between parallel syntactic structures is high because the projected spans represent the parts that synchronize between the languages without conflict. The source-language training sentences are sorted based on r_1. We select top-ranked unique source-language sentences with high r_1 as the data for the processing described in Sections 4.3 to 4.5. In our translation experiments in Section 6, we selected the top 0.1 million unique source-language sentences.

4.3. Producing Probabilistic Models for Parsing

The projected structures are usually partial structures. As full binary tree structures are required for learning our preordering model, we produce probabilistic models for parsing source-language sentences to produce full binary tree structures.

We will now discuss in detail our method for producing probabilistic models for parsing. The inputs are a source-language sentence F and projected partial syntactic structures of F, described in Section 4.1. The following task characteristics enable the use of

[2]The number of spans in a full binary tree is the number of words in a sentence minus one.

a simple model to produce full binary tree structures. (i) Partial structures are given. (ii) The set of phrase labels is predefined. We also predefine the number of types of POS tags. POS tags for each word are induced automatically.[3]

We build a probabilistic context-free grammar (CFG) model for parsing source-language sentences. We use the Pitman-Yor process (PY) [Pitman and Yor 1997][4] to build the model because it has a "rich-get-richer" property[5] that suits the process of learning a model from partially annotated structures. This is because information contained in annotated structures can be used[6] to infer structures that are not annotated. We also build a probabilistic model for unsupervised POS tagging using the Pitman-Yor process.

A probabilistic CFG is defined by the 4-tuple $G = (\mathcal{F}, V, S, \mathcal{R})$, where $\mathcal{F}$ is the set of terminals, which are source-language words in the training data, V is the set of nonterminals, $S \in V$ is a designated start symbol, and $\mathcal{R}$ is a set of rules. A CFG rule $x \rightarrow \alpha \in \mathcal{R}$ used in this process consists of $x \in V$ and α, which consists of two elements of V. V is defined as $V = \mathcal{L} \cup \mathcal{T}$, where $\mathcal{L}$ is the set of phrase labels for the target-language syntax, $\mathcal{T} = \{1, 2, ..., |\mathcal{T}|\}$ is the set of source-language POS tags represented by numbers, where $|\mathcal{T}|$ is the number of POS tag types, and $\mathcal{L} \cap \mathcal{T} = \varnothing$. Let $f \in \mathcal{F}$ be a source-language word and $F = f_1 f_2 .. f_m$. The probability of a derivation tree D is defined as the product of the probabilities of its component CFG rules and the probabilities of words given their POS tags, as follows.

$$P(D) = \prod_{x \rightarrow \alpha \in \mathcal{R}} P(\alpha|x)^{c(x \rightarrow \alpha, D)} \prod_{i=1}^{m} P(f_i|t_i), \tag{1}$$

where $c(x \rightarrow \alpha, D)$ is the number of times $x \rightarrow \alpha$ is used for the derivation D, $P(\alpha|x)$ is the probability of generating α given its root phrase label x, $t \in \mathcal{T}$ is a POS tag, index i of t indicates the position in F, and $P(f|t)$ is the probability of generating f given its POS tag t. The designated phrase label, S, is used for the phrase label of the root node of a tree.

Our PY models represent probability distributions over CFG rules or source-language words, as follows.

$$P(\alpha|x) \sim \mathrm{PY}_x(d_{\mathrm{cfg}}, \theta_{\mathrm{cfg}}, P_{\mathrm{base}}(\alpha|x)) \text{ and}$$
$$P(f|t) \sim \mathrm{PY}_t(d_{\mathrm{tag}}, \theta_{\mathrm{tag}}, P_{\mathrm{base}}(f|t)),$$

where d_{cfg}, θ_{cfg}, d_{tag}, and θ_{tag} are hyperparameters for the PY models. The hyperparameters are optimized via the auxiliary variable technique [Teh 2006].[7]

[3]POS tags are also thought to be projectable. However, some POS tags cannot be projected. For an example regarding translation between English and Japanese, determiners exist in English but do not exist in Japanese, and post positions exist in Japanese but not in English. In addition, a method that does not project POS tags is simpler than a method that projects POS tags.

[4]Readers unfamiliar with PY can refer to Teh [2006] for a detailed description and estimation method for PY.

[5]When one observation is sampled, then the probability of the posterior distribution for that observation becomes larger than that of the prior distribution.

[6]If a CFG rule appears many times in the annotated structures and has been sampled many times, then it is highly likely that the CFG rule will be sampled as a partial structure that is not annotated.

[7]We put a prior distribution of Beta$(1, 1)$ on d_{cfg} and d_{tag} and a prior distribution of Gamma$(1, 1)$ on θ_{cfg} and θ_{tag}.

The backoff probability distributions, $P_{\text{base}}(\alpha|x)$ and $P_{\text{base}}(f|t)$, are uniform, as follows.

$$P_{\text{base}}(\alpha|x) = \frac{1}{|V|^2} \text{ and}$$
$$P_{\text{base}}(f|t) = \frac{1}{|\mathcal{F}|},$$

where $|V|$ is the number of nonterminal types and $|\mathcal{F}|$ is the lexicon size of the source-language words in the training data. As our CFG rule has two leaf nodes, the number of pair nonterminal node types is $|V|^2$.

Sampling for building the distributions is conducted according to Equation (1) with the following constraints. When projected spans are present, we constrain the sampling such that only the derivation trees that do not conflict with the projected spans are sampled. Here, the conflict is that both a subtree span in the tree derivation and a projected span partially overlap each other. When there is an ambiguity in a projected span, which comprises the minimal projected span and any number of adjacent unaligned word(s), the constraints are as follows. If a sample (a *subtree* span in the tree derivation) does not conflict with the minimal projected span, then the minimal projected span is used as the constraint for the sample. Otherwise, the whole span (the minimal projected span and its adjacent unaligned word(s)) is used as the constraint for the sample. When the projected phrase label for a subtree span in a derivation tree is present, we constrain the sampling such that only the projected phrase label is sampled.

We use a sentence-level blocked Gibbs sampler based on a dynamic programming algorithm [Johnson et al. 2007]. The sampler consists of the following two steps: for each sentence, (1) inside probabilities [Lari and Young 1991] are calculated from the bottom up using the CYK algorithm, and (2) a tree is sampled from the top down according to the inside probabilities for each CFG rule. In the first step, when we calculate the inside probabilities for each phrase label in each cell of the triangular table of the CYK algorithm and save the inside probabilities, we also save the inside probabilities for each CFG rule. In the second step, we sample a CFG rule according to the inside probabilities for the CFG rules in each cell from the top down. To reduce computational costs, we only use N-best POS tags for each word when the inside probabilities are calculated. In our experiment in Section 6, we used five-best POS tags for each word.

The computational complexity of producing probabilistic models for parsing is linearly proportional to the number of training sentences when the data properties are identical except for the data size. The computational cost depends on the amount of nonconstrained parts in the data. When the amount of nonconstrained parts becomes larger, the computational cost becomes larger.

4.4. Parsing to Produce Full Binary Tree Structures

After the probability distributions of the PY models are built, we parse the source-language sentences to produce full binary tree structures that are maximally synchronized with the corresponding target-language structures. The parsing complements deficiencies and resolves ambiguities in the projected partial structures. The deficiencies are insufficient spans or phrase labels in the projected spans and labels to construct a full binary tree structure. The ambiguities of spans are shown as horizontal dotted lines in Figure 3, which cover nonaligned words adjacent to minimal projected spans. We calculate the maximum likelihood full binary tree structures based on the CYK algorithm within the constraints of the minimal projected spans and their phrase labels, using the produced probabilistic CFG model and the produced probabilistic

model for unsupervised POS tagging. The probability for a derivation tree is calculated using Equation (1). The constraints are the same constraints used for sampling when building the probabilistic models. The resulting full binary tree structures comprise the phrase labels of the target-language syntax. An example of the production of a full binary tree structure is shown in Figure 3.

Note that when the full binary trees are generated, all of the minimal projected spans are not necessarily included in the full binary trees if the projected spans have ambiguities. If nonaligned words are located adjacent to the projected spans, then there may be cases in which some minimal projected spans are not included in the full binary tree. For example, when a minimal projected span is $(f_1 f_2) f_3$ and f_3 is a nonaligned word where parentheses denote a span, a full binary tree may be $(f_1 (f_2 f_3))$. This tree does not include the minimal projected span because there are ambiguities in the span when a nonaligned word is adjacent to the span, as explained in Section 4.1.

4.5. Learning the Preordering Model

Learning of our preordering model uses the full binary tree structures of source-language sentences and word alignments.

The preordering model is a model based on two fundamental frameworks [Goto et al. 2013b]: (i) parsing using probabilistic CFG and (ii) the inversion transduction grammar (ITG) [Wu 1997]. In this article, the model combining (i) and (ii) is called the *ITG parsing model* and parsing using ITG is called *ITG parsing*. We use the ITG parsing model for preordering while Goto et al. [2013b] used this model for postordering.

To obtain the training data for the preordering model, we first obtain the reordered structure that produces the word order of F' most similar to the word order of the corresponding E using their word alignments. Figure 4 shows an example of the tree structure of F' calculated from the tree structure of F and word alignments. Reordering is conducted by swapping child nodes in the binary tree structure of F so that Kendall's τ is maximized between F' and E.[8] For each node, we decide whether its child nodes are swapped or not. This decision is made deterministically from the bottom up. The algorithm of this maximization can be expressed as $O(m^2 \log m)$ in complexity, where m is a sentence length. This is because the computational complexity of Kendall's τ can be expressed as $O(m \log m)$ [Knight 1966] for each node in a binary tree. When the scores for candidates are the same[9], we retain the original order.

The nodes whose child nodes are swapped to transform F into F' are then annotated with an "_SW" suffix (indicating "swap/inversion"), and other nodes with two child nodes are annotated with an "_ST" suffix (indicating "straight") in the binary tree for F. Figure 5 shows an example of F and its binary tree structure annotated with the _ST and _SW suffixes. The resulting binary tree syntactic structure of F is augmented with straight or swap/inversion suffixes, which can be regarded as a derivation of ITG between F and F'.

Thus, an ITG model can be learned from the binary tree structures using a probabilistic CFG learning algorithm. This learned model is the ITG parsing model. In this study, we use the state split probabilistic CFG [Petrov et al. 2006] for learning the ITG parsing model. The learned ITG parsing model is our preordering model.

5. PREORDERING SENTENCES

This section explains how to preorder input sentences and training sentences.

[8]Spearman's ρ will also work.

[9]The word order of nonaligned words does not affect Kendall's τ.

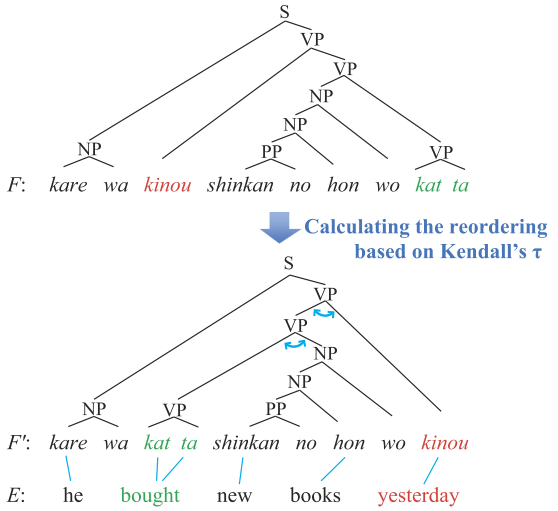


Fig. 4. Example of calculation of the reordering from F to F' based on Kendall's τ.

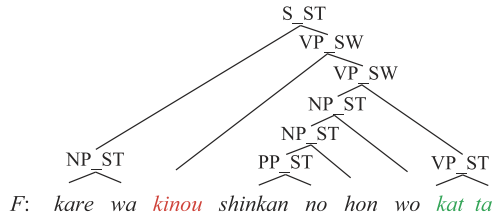


Fig. 5. Example of F and its binary tree structure annotated with _ST and _SW suffixes.

5.1. Preordering Input Sentences

Input sentences are preordered using the ITG parsing model described in Section 4.5. The preordering process is shown in Figure 6. An input sentence F is parsed using the ITG parsing model.[10] When F is parsed, the reordered structure for F' is jointly identified based on ITG. Each nonterminal node of a phrase label in the tree derivation is augmented by either an "_ST" suffix or an "_SW" suffix. The word order for F' is determined by the binary tree derivation with the suffixes of the nonterminal nodes. We swap the child nodes of the nodes augmented with the "_SW" suffix in the binary tree derivation to produce F'.

5.2. Preordering the Training Sentences

After transforming the F of an input sentence into F', we use a phrase-based SMT to translate F' into E. Therefore, a phrase-based SMT requires parallel F' and E sentences to train its translation model. Now, we will explain how to produce F' for the parallel sentences for training the SMT translation model.

If F' in the training data is produced using the same method as for preordering input sentences, then the word order of F' in the training data will be consistent with the word order of the preordered input sentences. However, the method for preordering input sentences is not always the best method for preordering the training data. This

[10]When there are unknown words, we estimate their POS tags using the function of the Berkeley parser [Petrov et al. 2006].

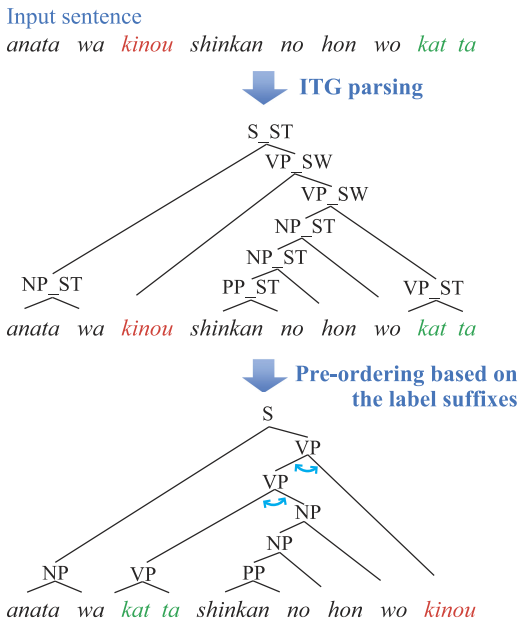


Fig. 6. Preordering an input sentence.

is because a corresponding E already exists in the training data. Thus, we also have to consider the consistency between F' and E in the training data. Methods that do not consider the consistency between F' and E will not be optimal.

It is important to consider the consistency between F' and E. The objective of preordering the training sentences is to build a phrase table. The phrase table is the SMT translation model, which consists of parallel phrase pairs between F' and E and their probabilities. When a pair of corresponding expressions in E and F' are both continuous, they can be extracted as a parallel phrase pair. A span in F that is projected by the method described in Section 4.1 indicates that the span in F and its corresponding span in E are both continuous. If the projected span in F is transformed into a noncontinuous expression in F' by preordering, then a parallel phrase pair for the noncontinuous expression in F' and the corresponding expression in E cannot be extracted as a phrase pair. Therefore, it is optimal that F be reordered into F' under the condition that this problem can be avoided, using (to the maximum possible extent) the same method used for preordering input sentences.

Thus, we preorder F in the training data into F' as follows. Partial syntactic structures are first projected onto the source-language sentences in the training data using the method described in Section 4.1. The source-language sentences are then parsed and reordered using the ITG parsing model (described in Section 5.1) within the constraints of the projected spans. The constraining method is the same as that used in Section 4.4.

6. EXPERIMENT

Our main goal is to translate text between two languages with widely different word orders, such as a SOV and SVO, with a high-quality target-language parser. Therefore, we conduct a Japanese-to-English (JE) translation to test the quality of translation from an SOV language to an SVO language. In addition, we conduct a Chinese-to-

English (CE) translation to test the quality of translation from an SVO language to another SVO language, as Chinese and English are more similar in terms of word order than Japanese and English. We investigate the efficacy of our method by comparing it with other methods. We used the patent data from the NTCIR-9 and NTCIR-10 Patent Machine Translation Tasks [Goto et al. 2011; Goto et al. 2013a] for the experiment.

6.1. Common Settings

The training data and the development data from the NTCIR-9 and NTCIR-10 are the same, but the test data are different. The JE training data consists of approximately 3.18 million sentence pairs and the CE training data consists of 1 million sentence pairs. The development data consists of 2,000 sentence pairs. There are 2,000 test sentences for the NTCIR-9 and 2,300 for the NTCIR-10. The reference data for each test sentence is a single reference translation.

We use Enju [Miyao and Tsujii 2008], which outputs head-binarized trees, to parse the English sentences in the training data. We applied a parsing customization for patent sentences [Isozaki et al. 2012]. MeCab[11] is used for Japanese segmentation, and the Stanford segmenter[12] is used for Chinese segmentation. We adjust the tokenization of alphanumeric characters in Japanese to be the same as for English.

The translation model is trained using sentences that are 40 words or less in length and English side sentences that could be parsed to produce binary syntactic tree structures. Approximately 2.06 million sentence pairs are used to train the translation model for JE. Approximately 0.40 million sentence pairs are used to train the translation model for CE. GIZA++ and grow-diag-final-and heuristics are used to obtain word alignments. To reduce word alignment errors, we remove the articles {a, an, the} in English and the particles {*ga*, *wo*, *wa*} in Japanese before performing word alignments, because these function words do not have corresponding words in the other languages between Japanese–English or Chinese–English. After word alignment, we restore the words that had been removed and shift the word alignment positions to the original word positions.

We use 5-gram language models with modified Kneser-Ney discounting [Chen and Goodman 1998] using the SRILM toolkit [Stolcke et al. 2011]. The language models are trained using the English sentences from the bilingual training data.

The SMT weighting parameters are tuned via MERT [Och 2003] using the development data. To stabilize the MERT results, we tune the parameters three times via MERT using the first half of the development data. We then select the SMT weighting parameter set that performs the best on the second half of the development data based on the BLEU scores from the three SMT weighting parameter sets.

6.2. Training and Settings for the Proposed Method

Next, we describe how the proposed method (PROPOSED) was performed. As the training data for our preordering model, we produce source-language full binary syntactic tree structures for 0.1 million source-language training sentences, which are selected using the method described in Section 4.2.[13] To produce the probabilistic CFG-model and the probabilistic model for unsupervised POS tagging, we use the Gibbs sampler for 100 iterations.[14] We use $|\mathcal{T}| = 50$, which employs the same number of word

[11] `http://mecab.sourceforge.net/`.

[12] `http://nlp.stanford.edu/software/segmenter.shtml`.

[13] We did not conduct experiments using larger training datasets because there would have been a very high computational cost in building probabilistic models for parsing.

[14] For JE, a single thread process ran for five days for 100 iterations on a Xeon processor E5-2680 2.70 GHz with 128GB memory.

classes used in the Moses default setting, where $|\mathcal{T}|$ is the number of POS tag types. The Berkeley parser [Petrov et al. 2006], which is an implementation of the state split probabilistic CFG-based parser, is used to train our preordering model and to parse using the preordering model. We perform six split-merge iterations as the same iteration of the parsing model for English [Petrov et al. 2006]. We use the phrase-based SMT system Moses [Koehn et al. 2007] to translate from F' into E with a distortion limit of 6, which limits the moves of phrases for word reordering to six or less words.

6.3. Training and Settings for the Comparison Methods

We used the following six comparison methods.

— Phrase-based SMT with lexicalized reordering models (PBMT$_L$) [Koehn et al. 2007].
— Hierarchical phrase-based SMT (HPBMT) [Chiang 2007].
— String-to-tree syntax-based SMT (SBMT) [Hoang et al. 2009].
— Phrase-based SMT with a distortion model (PBMT$_D$) [Goto et al. 2014].
— Preordering using a source-language dependency parser (SRCDEP) [Genzel 2010].[15]
— Preordering without using a parser (LADER) [Neubig et al. 2012].[16]

We use Moses [Hoang et al. 2009; Koehn et al. 2007] for PBMT$_L$, HPBMT, SBMT, SRCDEP, and LADER. We use an in-house standard phrase-based SMT decoder compatible with the Moses decoder with a distortion model [Goto et al. 2014] for PBMT$_D$.

For PBMT$_L$, we use the MSD bidirectional lexicalized reordering models [Koehn et al. 2005], which are built using all of the data used to build the translation model.

The distortion models for PBMT$_D$ are trained using 0.2 million source-language sentences[17] from the data used to build the translation model. This setting is the same as that in the experiments by Goto et al. [2014]. For PBMT$_D$, we use source-language POS tags produced by MeCab for Japanese and the Stanford tagger[18] for Chinese.

SRCDEP requires a source-language dependency parser. We use CaboCha[19] [Kudo and Matsumoto 2002] and POS tags produced by MeCab to obtain Japanese dependency structures[20] and use the Stanford parser[21] and POS tags produced by the Stanford tagger to obtain Stanford dependencies for Chinese [Chang et al. 2009]. Note that there are publicly available Japanese dependency parsers but there are no publicly available Japanese constituency parsers. The preordering rules of SRCDEP are built using all of the data used to build the translation model.

[15]There are three variations of the metrics for selecting rules. We implement variant 1 (optimizing crossing score), which achieved the best score for JE translation in the three variations in a study by Genzel [2010].
[16]We use the lader implementation available at `http://www.phontron.com/lader/`.
[17]The JE data is sorted in chronological order. The CE data is sorted at random. The last 0.2 million sentences of each data are used.
[18]`http://nlp.stanford.edu/software/tagger.shtml`.
[19]`https://code.google.com/p/cabocha/`.
[20]The CaboCha parser does not output word-based dependencies, but segment-based dependencies. Each segment, which is called a *bunsetsu*, comprises at least one content word, with or without its subsequent function words. We convert the segment-based dependencies to word-based dependencies as follows: when a punctuation mark is included in a segment, the segment is split into a segment without the punctuation mark and a segment consisting only of the punctuation mark. Each word, with the exception of the last word in a segment, depends on (modifies) the adjacent word to the right. The last word in a segment depends on the headword of the parent (modified) segment. The headword in a segment was the last content word in the segment.

The CaboCha parser does not output dependency relations. We add dependency relations to the word-based dependencies as follows: when the last word in a segment is a particle, we use the particle as the dependency relation between the word and its parent (modified) word because particles are case markers in many cases in Japanese. For other words, we use "none" as their dependency relations to their parent words.
[21]`http://nlp.stanford.edu/software/lex-parser.shtml`.

Table II. Japanese-English Evaluation Results

	Parser		Pre-	NTCIR-9		NTCIR-10	
	Source	Target	ordering	RIBES	BLEU	RIBES	BLEU
PBMT$_{L\text{-}4}$				65.48	26.73	65.53	27.44
PBMT$_{L\text{-}20}$				68.79	30.92	68.30	31.07
HPBMT				70.11	30.29	69.69	30.77
SBMT		✓		72.54	31.94	71.32	32.40
PBMT$_D$				73.54	33.14	72.23	33.87
SRCDEP	✓		✓	71.88	29.23	71.20	29.40
LADER			✓	74.31	32.98	73.98	33.90
PROPOSED		✓	✓	**76.35**	**33.83**	**75.81**	**34.90**

Table III. Chinese-English Evaluation Results

	Parser		Pre-	NTCIR-9		NTCIR-10	
	Source	Target	ordering	RIBES	BLEU	RIBES	BLEU
PBMT$_{L\text{-}4}$				75.02	29.22	74.24	30.65
PBMT$_{L\text{-}10}$				76.11	31.20	75.41	32.34
HPBMT				77.68	32.39	77.45	33.61
SBMT		✓		78.44	32.47	77.68	33.90
PBMT$_D$				77.98	33.03	77.48	34.28
SRCDEP	✓		✓	76.88	28.85	76.14	29.36
LADER			✓	78.18	30.80	77.06	31.12
PROPOSED		✓	✓	**81.61**	**35.16**	**81.05**	**36.22**

The preordering models for LADER are trained using the same 0.1 million source-language sentences and their word alignments as the training data for the preordering models of PROPOSED. We use source-language word classes produced by the Moses toolkit. Note that while the LADER preordering method does not use a parser, the training data for LADER is selected using a target-language parser. We perform 100 iterations to train the LADER preordering model.[22]

For PBMT$_L$, we use distortion limits of 4 or 20 for JE translation and distortion limits of 4 or 10 for CE translation, because a limit of 20 is the best for JE translation and a limit of 10 is the best for CE translation among 10, 20, 30, and ∞, as per studies by Goto et al. [2014] and because Genzel [2010] uses a baseline phrase-based SMT capable of local reordering of up to 4 words. To distinguish between the distortion limits for PBMT$_L$, we indicate the distortion limit as a subscript of PBMT$_L$, such as PBMT$_{L\text{-}20}$ for a distortion limit of 20. For PBMT$_D$, a distortion limit of 20 is used for JE translation and a distortion limit of 10 is used for CE translation. An unlimited max-chart-span is used for HPBMT and SBMT and a distortion limit of 6 is used for the preordering methods of SRCDEP and LADER. The default values are used for the other system parameters.

6.4. Results and Discussion

We evaluate the translation quality based on the case-insensitive automatic evaluation scores from the BLEU-4 [Papineni et al. 2002] and RIBES v1.01 [Isozaki et al. 2010]. RIBES is an automatic evaluation measure based on word-order correlation coefficients between reference sentences and translation outputs. Our main results for

[22]We also tested 200 iterations for JE translation and found that the results with 200 iterations did not improve when compared with the results for 100 iterations.

JE translation are presented in Table II and those for CE translation are presented in Table III. In these tables, check marks indicate usage for that method. Bold numbers indicate that values are not significantly lower than the best result (i.e., nonbold numbers indicate values that are significantly lower than the best result) in each test set and in each evaluation measure.[23] To assess this, we used the bootstrap resampling test at a significance level of $\alpha = 0.01$ [Koehn 2004].

PROPOSED achieved the best scores for both RIBES and BLEU in both the NTCIR-9 and NTCIR-10 datasets, and for both JE and CE translation. Because RIBES is sensitive to global word order and BLEU is sensitive to local word order, this confirms the efficacy of PROPOSED for both global and local word ordering.

Now, we compare the effects of the differences in the approaches. First, we compare our method with three existing methods that do not use a parser and conduct word selection and reordering jointly. For both the NTCIR-9 and NTCIR-10 results for JE translation, the RIBES and BLEU scores for PROPOSED are higher than those for the standard phrase-based SMT (PBMT$_{L\text{-}20}$), the hierarchical phrase-based SMT (HPBMT), and the phrase-based SMT with a recent distortion model (PBMT$_D$). These results confirm that preordering is effective compared with these methods, which do not use a parser and conduct word selection and reordering simultaneously, for JE patent translation. The tendencies of the CE translation results are the same as those of the JE translation results. These results confirm that preordering is also effective for CE patent translation.

Next, we compare our method with an existing method that uses a target-language syntactic parser, SBMT. The required resources are the same as those for PROPOSED. For both the NTCIR-9 and NTCIR-10 results for JE translation, the RIBES and BLEU scores for PROPOSED are higher than those for the string-to-tree syntax-based SMT (SBMT). These results confirm that preordering is effective compared with a method that uses target-language syntactic structures and conducts word selection and reordering simultaneously for JE patent translation. The tendencies of the CE translation results are also the same as those of the JE translation results.

We then compare our method with an existing method using a source-language dependency parser, SRCDEP. For both the NTCIR-9 and NTCIR-10 results for JE translation, the RIBES and BLEU scores for PROPOSED are higher than those for SRCDEP. These results confirm that our method is effective compared with a method that uses a source-language dependency parser [Genzel 2010] for JE patent translation. The tendencies of the CE translation results are the same as those of the JE translation results.

We confirm the effects of SRCDEP for JE (i.e., SOV to SVO) translation.[24] SRCDEP produces BLEU scores that are about two BLEU points higher than those for PBMT$_{L\text{-}4}$. These results are consistent with the experimental results of Genzel [2010]. Genzel [2010] compares their method with their baseline phrase-based SMT, which is capable of local reordering of up to four words. Although SRCDEP produces better BLEU scores than those for PBMT$_{L\text{-}4}$ and better RIBES scores than those for PBMT$_{L\text{-}4}$ and PBMT$_{L\text{-}20}$, the BLEU scores for SRCDEP are lower than those for PBMT$_{L\text{-}20}$. This indicates that even if a source-language dependency parser is used, it is not easy to

[23] We use this indication method because it can clarify the results of a hypothesis test using one result and many baseline results.

[24] Since Genzel [2010] reported the results of translations from English (an SVO language) to SOV or VSO languages, including Japanese, and did not report the results of a translation between English and Chinese (an SVO language to an SVO language), we discuss SRCDEP for JE translation.

Table IV. Effects of the Sentence Selection Method (JE Translation)

	NTCIR-9		NTCIR-10	
	RIBES	BLEU	RIBES	BLEU
LADER without our sentence selection	72.33	32.30	70.96	33.07
LADER with our sentence selection	74.31	32.98	73.98	33.90

improve JE translation quality by preordering.[25] One of the reasons that SRCDEP is unable to achieve scores on par with PROPOSED is thought to be because when SRCDEP changes the order of child nodes, the reordering rules consider only the local information. Reordering, however, should consider sentence-level consistency. For example, an SOV sentence in Japanese should be reordered into an SVO sentence for JE translation. However, when the subject in a sentence is omitted in Japanese, an OV sentence in Japanese should not be reordered into a VO sentence. This is because such sentences are usually translated into sentences in the passive voice, and the objects in Japanese become subjects in the translated sentences. Because SRCDEP preordering rules only consider local information, a rule is unable to handle the difference between SOV and OV when the rule does not consider S, such as when swapping O and V. In contrast, PROPOSED considers sentence-level consistency.

Finally, we compare our method with an existing preordering method that does not use a syntactic parser, LADER. For both the NTCIR-9 and NTCIR-10 results for JE translation, the RIBES and BLEU scores for PROPOSED are higher than those for LADER.[26] These results confirm that syntactic structures are effective for preordering in JE patent translation.[27] The tendencies of the CE translation results are also the same as those of the JE translation results.

Here, we confirm the effects of our method for selecting the training sentences described in Section 4.2 for JE translation. Because the sentence selection method is an indispensable element of our method, we compare LADER with our sentence selection method, which is LADER in Table II, to LADER without our sentence selection method. We used 0.1 million source-language sentences from the training data as the training data for the preordering model of LADER without our sentence selection method. The size of the 0.1 million sentences is the same as the size of the training data for the

[25]There are also systems with preordering methods that use a source-language dependency parser in the Japanese-to-English translation subtasks at NTCIR-10 and NTCIR-7.

At NTCIR-10, there was one system (name: JEPREORDER, ID: NTITI-je-2) with a source syntax-based preordering method that used manually-produced preordering rules and a Japanese dependency parser with a case structure analyzer [Sudoh et al. 2013]. Compared with the baseline hierarchical phrase-based SMT system (ID: BASELINE1-1) at NTCIR-10, the BLEU score for JEPREORDER is higher than that of the baseline system, but the RIBES score is not better than that of the baseline system in Table 1 in Sudoh et al. [2013].

At NTCIR-7, there was one system (ID: MIT (2)) with a source syntax-based preordering method that used manually produced preordering rules and a Japanese dependency parser [Katz-Brown and Collins 2008]. This system was unable to produce a BLEU score that was better than that of the baseline phrase-based SMT system at NTCIR-7.

[26]Note that although LADER works without a syntactic parser, the scores for LADER in Table II could not be achieved without a syntactic parser, because a syntactic parser is used in the selection process of the training data for the preordering model of LADER. The results for cases when our sentence selection method is not applied for LADER are shown later in this section. We use the same training data for the preordering model of LADER as for the preordering model of PROPOSED to ensure a fair comparison.

[27]There was also a system with a preordering method that does not require a parser in the Japanese-to-English translation subtask at NTCIR-9. The system of the NAIST group [Kondo et al. 2011] used a preordering method [Tromble and Eisner 2009] that learned a preordering model automatically without requiring a parser. This system was unable to produce a BLEU score that was better than those for the baseline systems of phrase-based SMT and hierarchical phrase-based SMT at NTCIR-9, although it could produce a RIBES score that was better than those for the baseline systems.

preordering model of LADER in Table II. The results are shown in Table IV. The RIBES and BLEU scores for LADER with our sentence selection method are higher than those for LADER without our sentence selection method. This comparison confirms the efficacy of our sentence selection method. This result also confirms that the learning of BTG from parallel sentences with highly synchronized parallel structures is effective compared with the learning from parallel sentences with less synchronized parallel structures.

Through the process described in Section 4.1, we assess the percentage of the spans in the source language that do not conflict in all of the spans that could be projected from the target language. As we explain in Section 4.1, to ensure that the projected structures can compose tree structures and consist solely of high-quality structures, we do not project any subtree spans in E when their corresponding spans in F conflict with one other. The nonconflict span rate in the source language is calculated by dividing the number of projected nonconflict spans in the source language by the number of spans in the source language that could be projected without consideration of conflict. The nonconflict span rates for the data selected to build the translation model are 0.782 for Japanese and 0.747 for Chinese.

We also check the average coverage rates of the projected spans, except for the sentence root spans,[28] in the process described in Section 4.1. Let r_2 be the coverage rate of the projected spans, except for the root span, for a source-language sentence. r_2 is calculated by dividing the number of projected spans, except for the sentence root span, by the number of words in a sentence minus two.[29,30] The average r_2 for the data selected to build the translation model (2.06 million Japanese sentences and 0.40 million Chinese sentences) is 0.560 for Japanese and 0.601 for Chinese. The average r_2 for the 0.1 million sentences selected via our sentence selection method (described in Section 4.2) is 0.856 for Japanese and 0.828 for Chinese.[31] With these projected partial structures, full binary tree structures were produced using the methods described in Sections 4.3, 4.4, and 5.2.[32]

In these experiments, we have not compared our method with postordering methods. However, for the same NTCIR-9 test data, the RIBES score (76.35) and the BLEU score (33.83) for PROPOSED are higher than the RIBES score (75.12) and the BLEU score (32.95) reported in Goto et al. [2013b], which were calculated in the same way as ours, for a postordering method [Goto et al. 2013b]. The postordering method of Goto et al. [2013b] used the same state split probabilistic CFG method for the ITG parsing model as our method for the ITG parsing model. In addition, PROPOSED has an advantage over the postordering methods of Sudoh et al. [2011b], Goto et al. [2013b], and Hayashi et al. [2013]. These postordering methods use manually-defined, high-quality preordering rules of head-finalization for translation from English to Japanese [Isozaki et al. 2012], so it is not easy to apply these methods to other language pairs. In contrast, PROPOSED does not require such manually-defined rules, and thus can be applied to other languages.

[28]As sentence root spans are obvious and do not need to be projected, we exclude them to investigate the percentage of spans that are projected.

[29]The number of spans in a full binary tree is the number of words in a sentence minus one. We subtract one from the number of spans to remove the sentence root span.

[30]r_2 can also be used instead of r_1 for sentence selection in Section 4.2.

[31]We also report average r_1. The average r_1 for the data selected to build the translation model is 0.574 for Japanese and 0.613 for Chinese. The average r_1 for the 0.1 million sentences is 0.864 for Japanese and 0.834 for Chinese.

[32]This does not mean that all of the minimal projected spans are included in the full binary trees. The reason for this is given in Section 4.4.

Table V. Evaluation Results for Parsing

	F_1 (CTB5-40)
[Jiang et al. 2011]	49.2*
Proposed method	**56.1**

Note: ∗ denotes "not our experiment."

6.5. Evaluation of Projection

To investigate the effects of our projection method, we compare the parsing quality produced by our method with that produced by the method of Jiang et al. [2011]. We use the same data and evaluation method as Jiang et al. [2011]. We use the same FBIS Chinese–English parallel corpus (LDC2003E14), which consists of 0.24 million sentence pairs, to obtain projected constituent structures. We evaluate our projected parser using the same test data as the subset of Chinese Treebank 5.0 (CTB 5.0; LDC2005T01), which consists of no more than 40 words after the removal of punctuation, just as in Jiang et al. [2011].

We use the same evaluation metric of unlabeled F_1 as Jiang et al. [2011], which is the harmonic mean of the unlabeled precision and recall. This is defined by Klein [2005, pp. 19–22]. The evaluation for unlabeled brackets differs slightly from the standard PARSEVAL metrics: multiplicity of brackets is ignored, brackets with a span of one are ignored, and bracket labels are ignored. Previous research [Jiang et al. 2011; Klein 2005, p. 16] removed punctuation before conducting the evaluations. Followed this, we remove words that have PU punctuation tags in CTB 5.0 after parsing.

We use our method, described in Sections 4.1 to 4.4, 6.1, and 6.2 to obtain projected constituent structures. As a result of the process described in Section 4.1, the nonconflict span rate in the source language for the training data of the FBIS corpus is 0.681. To reduce computational costs, we change one of the settings described in Section 6.2. To produce a probabilistic CFG model and full binary trees, we select the top 50,000 unique source-language sentences from the FBIS corpus, whereas we selected the top 0.1 million unique source-language sentences in Section 6.2.[33] The average coverage rate (r_2) of the projected spans, except for the sentence root spans, for the 50,000 sentences selected to produce full binary tree structures is 0.795. We use the Berkeley parser, which was also used by Jiang et al. [2011] for the same purpose, to build the parsing model from the projected constituent structures and to parse the test data.

Jiang et al. [2011] uses the gold POS tags from CTB 5.0 for parsing and a supervised Chinese POS tagger for tagging the FBIS corpus. In contrast, we do not use the gold POS tags from CTB 5.0 or a supervised Chinese POS tagger. Therefore, a comparison of our method with that of Jiang et al. [2011] would be unfair.

The evaluation results are given in Table V. Although our method does not require source-language POS tags, our method produced an F_1 higher than that of Jiang et al. [2011]. This confirms the efficacy of our projection method.

The quality of projected spans is affected by the quality of word alignments between languages as well as the quality of target-language parse trees. To check the quality of spans projected by the method described in Section 4.1, we use the English translation with annotations[34] (LDC2007T02) of a part of CTB 5.0 and the corresponding Chinese sentences in CTB 5.0. From the data, we extract parallel sentence pairs that have: (1) a one-to-one sentence correspondence, (2) sentence lengths of 40 words or less, and

[33]The average r_2 of the top 0.1 million sentences in the FBIS corpus is 0.668, which is lower than that of the NTCIR-9/10 data (0.856 for Japanese and 0.828 for Chinese). A lower rate increases the number of parse tree candidates and also the computational costs.

[34]We use the annotations to count the number of English sentences corresponding to each Chinese sentence.

(3) English side sentences that could be parsed to produce binary syntactic tree structures. The extracted Chinese–English parallel sentence pairs are called parallel CTB in this article. The parallel CTB comprise 2,477 sentence pairs. We obtain word alignments using the parallel CTB and the FBIS corpus simultaneously. Then, we project spans using our method described in Section 4.1. We check the quality of the minimal projected spans for the parallel CTB. Error spans, which are also called cross brackets, are the projected spans that conflict with subtree spans of CTB 5.0. The conflict is that a projected span and a subtree span in the syntactic tree of CTB 5.0 for a sentence partially overlap each other. The error rate of the minimal projected spans is 0.217, which is calculated by dividing the number of error spans by the number of projected spans, except for the root spans.[35] For the Chinese sentences from the parallel CTB, the nonconflict span rate is 0.689, and the average coverage rate (r_2) of the projected spans, except for the root spans, is 0.608.

7. CONCLUSION

We have presented a preordering method that uses a target-language parser to process syntactic structures without a source-language parser. This is achieved by projecting source-language syntactic structures from the corresponding target-language constituency structures, as well as learning an ITG-based parsing model for the source-language using the projected syntactic structures. Our method, which is based on cross-language syntactic projection and sentence selection, facilitates the learning of ITG by producing highly-synchronized parallel syntactic structures. In the experiments on Japanese-to-English and Chinese-to-English patent translation, our method is significantly better in terms of translation quality as measured by both RIBES and BLEU when compared with phrase-based SMT, hierarchical phrase-based SMT, string-to-tree syntax-based SMT, an existing preordering method that does not use a parser, and an existing preordering method that uses a source-language dependency parser. As RIBES is sensitive to global word order and BLEU is sensitive to local word order, we conclude that our proposed method is better than the compared methods in terms of global and local word ordering. We also confirm the efficacy of our projection method compared with an existing projection method for constituent structures using the FBIS corpus and Chinese Treebank 5.0. Future work will involve cooperating with a source-language parser when one is available.

ACKNOWLEDGMENTS

I. Goto would like to thank Chenhui Chu for providing sentence alignments between LDC2007T02 and CTB5.0. We would like to thank the three anonymous reviewers for their comments which substantially improved the article.

REFERENCES

A. V. Aho and J. D. Ullman. 1969. Syntax directed translations and the pushdown assembler. *J. Comput. Syst. Sci. 3*, 1, 37–56. DOI: http://dx.doi.org/10.1016/S0022-0000(69)80006-1.

Ibrahim Badr, Rabih Zbib, and James Glass. 2009. Syntactic phrase reordering for English-to-Arabic statistical machine translation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL'09)*. Association for Computational Linguistics, 86–93. http://www.aclweb.org/anthology/E09-1011.

Pi-Chuan Chang, Huihsin Tseng, Dan Jurafsky, and Christopher D. Manning. 2009. Discriminative reordering with Chinese grammatical relations features. In *Proceedings of the 3rd Workshop on Syntax and*

[35]The minimal projected spans, with the exception of the error spans, do not always match the correct spans because there may be ambiguities in the projected spans and the tree structures of CTB5.0 are not binary trees.

Structure in Statistical Translation (SSST-3) at NAACL HLT. Association for Computational Linguistics, 51–59. http://www.aclweb.org/anthology/W09-2307.

Stanley F. Chen and Joshua T. Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98. Computer Science Group, Harvard University.

David Chiang. 2007. Hierarchical phrase-based translation. *Comput. Linguist. 33*, 2, 201–228.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. Association for Computational Linguistics, 531–540. DOI: http://dx.doi.org/10.3115/1219840.1219906.

John DeNero and Jakob Uszkoreit. 2011. Inducing sentence structure from parallel corpora for reordering. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 193–203. http://www.aclweb.org/anthology/D11-1018.

Chris Dyer and Philip Resnik. 2010. Context-free reordering, finite-state translation. In *Human Language Technologies: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 858–866. http://www.aclweb.org/anthology/N10-1128.

Niyu Ge. 2010. A direct syntax-driven reordering model for phrase-based machine translation. In *Human Language Technologies: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 849–857. http://www.aclweb.org/anthology/N10-1127.

Dmitriy Genzel. 2010. Automatically learning source-side reordering rules for large scale machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*, 376–384. http://www.aclweb.org/anthology/C10-1043.

Isao Goto, Bin Lu, Ka Po Chow, Eiichiro Sumita, and Benjamin K. Tsou. 2011. Overview of the patent machine translation task at the NTCIR-9 Workshop. In *Proceedings of NTCIR-9*. 559–578.

Isao Goto, Ka Po Chow, Bin Lu, Eiichiro Sumita, and Benjamin K. Tsou. 2013a. Overview of the patent machine translation task at the NTCIR-10 Workshop. In *Proceedings of NTCIR-10*. 260–286.

Isao Goto, Masao Utiyama, and Eiichiro Sumita. 2013b. Post-ordering by parsing with ITG for Japanese-English statistical machine translation. *ACM Trans. Asian Lang. Inf. Process. 12*, 4, Article 17, 22 pages. DOI: http://dx.doi.org/10.1145/2518100.

Isao Goto, Masao Utiyama, Eiichiro Sumita, Akihiro Tamura, and Sadao Kurohashi. 2014. Distortion model based on word sequence labeling for statistical machine translation. *ACM Trans. Asian Lang. Inf. Process. 13*, 1, Article 2, 21 pages. DOI: http://dx.doi.org/10.1145/2537128.

Nizar Habash. 2007. Syntactic preprocessing for statistical machine translation. In *Proceedings of the Machine Translation Summit XI*. 215–222.

Katsuhiko Hayashi, Katsuhito Sudoh, Hajime Tsukada, Jun Suzuki, and Masaaki Nagata. 2013. Shift-reduce word reordering for machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1382–1386. http://www.aclweb.org/anthology/D13-1139.

Hieu Hoang, Philipp Koehn, and Adam Lopez. 2009. A unified framework for phrase-based, hierarchical, and syntax-based statistical machine translation. In *Proceedings of the International Workshop on Spoken Language Translation*. 152–159.

Sho Hoshino, Yusuke Miyao, Katsuhito Sudoh, and Masaaki Nagata. 2013. Two-stage preordering for Japanese-to-English statistical machine translation. In *Proceedings of the 6th International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, 1062–1066. http://www.aclweb.org/anthology/I13-1147.

Rebecca Hwa, Philip Resnik, Amy Weinberg, Clara Cabezas, and Okan Kolak. 2005. Bootstrapping parsers via syntactic projection across parallel texts. *Nat. Lang. Eng. 11*, 3, 311–325. DOI: http://dx.doi.org/10.1017/S1351324905003840.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. 944–952.

Hideki Isozaki, Katsuhito Sudoh, Hajime Tsukada, and Kevin Duh. 2012. HPSG-based preprocessing for English-to-Japanese translation. *ACM Trans. Asian Lang. Inf. Process. 11*, 3, Article 8, 16 pages. DOI: http://dx.doi.org/10.1145/2334801.2334802.

Wenbin Jiang, Qun Liu, and Yajuan Lv. 2011. Relaxed cross-lingual projection of constituent syntax. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1192–1201. http://www.aclweb.org/anthology/D11-1110.

Mark Johnson, Thomas Griffiths, and Sharon Goldwater. 2007. Bayesian inference for PCFGs via Markov Chain Monte Carlo. In *Human Language Technologies: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 139–146. http://www.aclweb.org/anthology/N/N07/N07-1018.

Jason Katz-Brown and Michael Collins. 2008. Syntactic reordering in preprocessing for Japanese → English translation: MIT system description for NTCIR-7 patent translation task. In *Proceedings of the NTCIO-7*. 409–414.

Mitesh M. Khapra, Ananthakrishnan Ramanathan, and Karthik Visweswariah. 2013. Improving reordering performance using higher order and structural features. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, 315–324. http://www.aclweb.org/anthology/N13-1032.

Dan Klein. 2005. The unsupervised learning of natural language structure. Ph.D. Dissertation, Stanford University.

William R. Knight. 1966. A computer method for calculating Kendall's tau with ungrouped data. *J. Amer. Statist. Assoc. 61*, 314. DOI: http://dx.doi.org/10.2307/2282833.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, Dekang Lin and Dekai Wu Eds. Association for Computational Linguistics, 388–395.

Philipp Koehn, Amittai Axelrod, Alexandra Birch Mayne, Chris Callison-Burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 IWSLT speech translation evaluation. In *Proceedings of the International Workshop on Spoken Language Translation*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Association for Computational Linguistics, 177–180. http://www.aclweb.org/anthology/P07-2045.

Shuhei Kondo, Mamoru Komachi, Yuji Matsumoto, Katsuhito Sudoh, Kevin Duh, and Hajime Tsukada. 2011. Learning of linear ordering problems and its application to J-E patent translation in NTCIR-9 PatentMT. In *Proceedings of NTCIR-9*. 641–645.

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analysis using cascaded chunking. In *Proceedings of the 6th Conference on Natural Language Learning (CoNLL'02) (COLING 2002 Post-Conference Workshops)*. 63–69.

K. Lari and S. J. Young. 1991. Applications of stochastic context-free grammars using the Inside-Outside algorithm. *Comput. Speech Lang. 5*, 3, 237–257. DOI: http://dx.doi.org/10.1016/0885-2308(91)90009-F.

Uri Lerner and Slav Petrov. 2013. Source-side classifier preordering for machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 513–523. http://www.aclweb.org/anthology/D13-1049.

Chi-Ho Li, Minghui Li, Dongdong Zhang, Mu Li, Ming Zhou, and Yi Guan. 2007. A probabilistic approach to syntax-based reordering for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, 720–727. http://www.aclweb.org/anthology/P07-1091.

Yusuke Miyao and Jun'ichi Tsujii. 2008. Feature forest models for probabilistic HPSG parsing. *Comput. Linguist. 34*, 1, 81–88.

Graham Neubig, Taro Watanabe, and Shinsuke Mori. 2012. Inducing a discriminative parser to optimize machine translation reordering. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 843–853. http://www.aclweb.org/anthology/D12-1077.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 160–167. DOI: http://dx.doi.org/10.3115/1075096.1075117.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. 311–318.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 433–440. DOI: http://dx.doi.org/10.3115/1220175.1220230.

Jim Pitman and Marc Yor. 1997. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *Ann. Probab. 25*, 2, 855–900.

Ananthakrishnan Ramanathan, Hegde, Jayprasad, Ritesh M. Shah, Pushpak Bhattacharyya, and Sasikumar M. 2008. Simple syntactic and morphological processing can help English-Hindi statistical machine translation. In *Proceedings of the 3rd International Joint Conference on Natural Language Processing*. 171–180.

Andreas Stolcke, Jing Zheng, Wen Wang, and Victor Abrash. 2011. SRILM at sixteen: Update and outlook. In *Proceedings of the IEEE Automatic Speech Recognition and Understanding Workshop*.

Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, Masaaki Nagata, Xianchao Wu, Takuya Matsuzaki, and Jun'ichi Tsujii. 2011a. NTT-UT statistical machine translation in NTCIR-9 PatentMT. In *Proceedings of NTCIR-9*. 585–592.

Katsuhito Sudoh, Jun Suzuki, Hajime Tsukada, and Masaaki Nagata. 2013. NTT-NII statistical machine translation for NTCIR-10 PatentMT. In *Proceedings of NTCIR-10*. 294–300.

Katsuhito Sudoh, Xianchao Wu, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011b. Post-ordering in statistical machine translation. In *Proceedings of the 13th Machine Translation Summit*. 316–323.

Yee Whye Teh. 2006. A Bayesian interpretation of interpolated Kneser-Ney. Technical Report TRA2/06. NUS School of Computing.

Christoph Tillman. 2004. A unigram orientation model for statistical machine translation. In *Proceedings of the HLT-NAACL'04* (Short Papers), Daniel Marcu, Susan Dumais, and Salim Roukos (Eds.), Association for Computational Linguistics, USA, 101–104.

Roy Tromble and Jason Eisner. 2009. Learning linear ordering problems for better translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1007–1016. http://www.aclweb.org/anthology/D/D09/D09-1105.

Karthik Visweswariah, Jiri Navratil, Jeffrey Sorensen, Vijil Chenthamarakshan, and Nandakishore Kambhatla. 2010. Syntax based reordering with automatically derived rules for improved statistical machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING'10)*. Coling 2010 Organizing Committee, 1119–1127. http://www.aclweb.org/anthology/C10-1126.

Karthik Visweswariah, Rajakrishnan Rajkumar, Ankur Gandhe, Ananthakrishnan Ramanathan, and Jiri Navratil. 2011. A word reordering model for improved machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 486–496. http://www.aclweb.org/anthology/D11-1045.

Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Association for Computational Linguistics, 737–745. http://www.aclweb.org/anthology/D/D07/D07-1077.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Comput. Linguist. 23*, 3, 377–403.

Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011a. Extracting preordering rules from chunk-based dependency trees for Japanese-to-English translation. In *Proceedings of the 13th Machine Translation Summit*. 300–307.

Xianchao Wu, Katsuhito Sudoh, Kevin Duh, Hajime Tsukada, and Masaaki Nagata. 2011b. Extracting preordering rules from predicate-argument structures. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, 29–37. http://www.aclweb.org/anthology/I11-1004.

Fei Xia and Michael McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of COLING'04*. 508–514.

Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a dependency parser to improve SMT for subject-object-verb languages. In *Human Language Technologies: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 245–253. http://www.aclweb.org/anthology/N/N09/N09-1028.

Kenji Yamada and Kevin Knight. 2001. A syntax-based statistical translation model. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 523–530. DOI: http://dx.doi.org/10.3115/1073012.1073079.

Received March 2014; revised September 2014; accepted November 2014

Machine Learning 1: 81–106, 1986

Induction of Decision Trees

J.R. QUINLAN (munnari!nswitgould.oz!quinlan@seismo.css.gov)
Centre for Advanced Computing Sciences, New South Wales Institute of Technology, Sydney 2007, Australia

(Received August 1, 1985)

Key words: classification, induction, decision trees, information theory, knowledge acquisition, expert systems

Abstract. The technology for building knowledge-based systems by inductive inference from examples has been demonstrated successfully in several practical applications. This paper summarizes an approach to synthesizing decision trees that has been used in a variety of systems, and it describes one such system, ID3, in detail. Results from recent studies show ways in which the methodology can be modified to deal with information that is noisy and/or incomplete. A reported shortcoming of the basic algorithm is discussed and two means of overcoming it are compared. The paper concludes with illustrations of current research directions.

1. Introduction

Since artificial intelligence first achieved recognition as a discipline in the mid 1950's, machine learning has been a central research area. Two reasons can be given for this prominence. The ability to learn is a hallmark of intelligent behavior, so any attempt to understand intelligence as a phenomenon must include an understanding of learning. More concretely, learning provides a potential methodology for building high-performance systems.

Research on learning is made up of diverse subfields. At one extreme there are adaptive systems that monitor their own performance and attempt to improve it by adjusting internal parameters. This approach, characteristic of a large proportion of the early learning work, produced self-improving programs for playing games (Samuel, 1967), balancing poles (Michie, 1982), solving problems (Quinlan, 1969) and many other domains. A quite different approach sees learning as the acquisition of structured knowledge in the form of concepts (Hunt, 1962; Winston, 1975), discrimination nets (Feigenbaum and Simon, 1963), or production rules (Buchanan, 1978).

The practical importance of machine learning of this latter kind has been underlin-

ed by the advent of knowledge-based expert systems. As their name suggests, these systems are powered by knowledge that is represented explicitly rather than being implicit in algorithms. The knowledge needed to drive the pioneering expert systems was codified through protracted interaction between a domain specialist and a knowledge engineer. While the typical rate of knowledge elucidation by this method is a few rules per man day, an expert system for a complex task may require hundreds or even thousands of such rules. It is obvious that the interview approach to knowledge acquisition cannot keep pace with the burgeoning demand for expert systems; Feigenbaum (1981) terms this the 'bottleneck' problem. This perception has stimulated the investigation of machine learning methods as a means of explicating knowledge (Michie, 1983).

This paper focusses on one microcosm of machine learning and on a family of learning systems that have been used to build knowledge-based systems of a simple kind. Section 2 outlines the features of this family and introduces its members. All these systems address the same task of inducing decision trees from examples. After a more complete specification of this task, one system (ID3) is described in detail in Section 4. Sections 5 and 6 present extensions to ID3 that enable it to cope with noisy and incomplete information. A review of a central facet of the induction algorithm reveals possible improvements that are set out in Section 7. The paper concludes with two novel initiatives that give some idea of the directions in which the family may grow.

2. The TDIDT family of learning systems

Carbonell, Michalski and Mitchell (1983) identify three principal dimensions along which machine learning systems can be classified:

- the underlying learning strategies used;
- the representation of knowledge acquired by the system; and
- the application domain of the system.

This paper is concerned with a family of learning systems that have strong common bonds in these dimensions.

Taking these features in reverse order, the *application domain* of these systems is not limited to any particular area of intellectual activity such as Chemistry or Chess; they can be applied to any such area. While they are thus general-purpose systems, the applications that they address all involve *classification*. The product of learning is a piece of procedural knowledge that can assign a hitherto-unseen object to one of a specified number of disjoint classes. Examples of classification tasks are:

1. the diagnosis of a medical condition from symptoms, in which the classes could be either the various disease states or the possible therapies;
2. determining the game-theoretic value of a chess position, with the classes *won for white*, *lost for white*, and *drawn*; and
3. deciding from atmospheric observations whether a severe thunderstorm is unlikely, possible or probable.

It might appear that classification tasks are only a minuscule subset of procedural tasks, but even activities such as robot planning can be recast as classification problems (Dechter and Michie, 1985).

The members of this family are sharply characterized by their *representation of acquired knowledge* as decision trees. This is a relatively simple knowledge formalism that lacks the expressive power of semantic networks or other first-order representations. As a consequence of this simplicity, the learning methodologies used in the TDIDT family are considerably less complex than those employed in systems that can express the results of their learning in a more powerful language. Nevertheless, it is still possible to generate knowledge in the form of decision trees that is capable of solving difficult problems of practical significance.

The *underlying strategy* is non-incremental learning from examples. The systems are presented with a set of cases relevant to a classification task and develop a decision tree from the top down, guided by frequency information in the examples but not by the particular order in which the examples are given. This contrasts with incremental methods such as that employed in MARVIN (Sammut, 1985), in which a dialog is carried on with an instructor to 'debug' partially correct concepts, and that used by Winston (1975), in which examples are analyzed one at a time, each producing a small change in the developing concept; in both of these systems, the order in which examples are presented is most important. The systems described here search for patterns in the given examples and so must be able to examine and re-examine all of them at many stages during learning. Other well-known programs that share this data-driven approach include BACON (Langley, Bradshaw and Simon, 1983) and INDUCE (Michalski, 1980).

In summary, then, the systems described here develop decision trees for classification tasks. These trees are constructed beginning with the root of the tree and proceeding down to its leaves. The family's palindromic name emphasizes that its members carry out the *T*op-*D*own *I*nduction of *D*ecision *T*rees.

The example objects from which a classification rule is developed are known only through their values of a set of properties or attributes, and the decision trees in turn are expressed in terms of these same attributes. The examples themselves can be assembled in two ways. They might come from an existing database that forms a history of observations, such as patient records in some area of medicine that have accumulated at a diagnosis center. Objects of this kind give a reliable statistical picture but, since they are not organized in any way, they may be redundant or omit

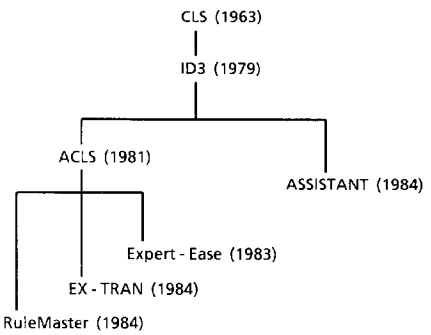


Figure 1. The TDIDT family tree.

uncommon cases that have not been encountered during the period of record-keeping. On the other hand, the objects might be a carefully culled set of tutorial examples prepared by a domain expert, each with some particular relevance to a complete and correct classification rule. The expert might take pains to avoid redundancy and to include examples of rare cases. While the family of systems will deal with collections of either kind in a satisfactory way, it should be mentioned that earlier TDIDT systems were designed with the 'historical record' approach in mind, but all systems described here are now often used with tutorial sets (Michie, 1985).

Figure 1 shows a family tree of the TDIDT systems. The patriarch of this family is Hunt's Concept Learning System framework (Hunt, Marin and Stone, 1966). CLS constructs a decision tree that attempts to minimize the cost of classifying an object. This cost has components of two types: the measurement cost of determining the value of property A exhibited by the object, and the misclassification cost of deciding that the object belongs to class J when its real class is K. CLS uses a lookahead strategy similar to minimax. At each stage, CLS explores the space of possible decision trees to a fixed depth, chooses an action to minimize cost in this limited space, then moves one level down in the tree. Depending on the depth of lookahead chosen, CLS can require a substantial amount of computation, but has been able to unearth subtle patterns in the objects shown to it.

ID3 (Quinlan, 1979, 1983a) is one of a series of programs developed from CLS in response to a challenging induction task posed by Donald Michie, viz. to decide from pattern-based features alone whether a particular chess position in the King-Rook vs King-Knight endgame is lost for the Knight's side in a fixed number of ply. A full description of ID3 appears in Section 4, so it is sufficient to note here that it embeds a tree-building method in an iterative outer shell, and abandons the cost-driven lookahead of CLS with an information-driven evaluation function.

ACLS (Paterson and Niblett, 1983) is a generalization of ID3. CLS and ID3 both require that each property used to describe objects has only values from a specified set. In addition to properties of this type, ACLS permits properties that have

unrestricted integer values. The capacity to deal with attributes of this kind has allowed ACLS to be applied to difficult tasks such as image recognition (Shepherd, 1983).

ASSISTANT (Kononenko, Bratko and Roskar, 1984) also acknowledges ID3 as its direct ancestor. It differs from ID3 in many ways, some of which are discussed in detail in later sections. ASSISTANT further generalizes on the integer-valued attributes of ACLS by permitting attributes with continuous (real) values. Rather than insisting that the classes be disjoint, ASSISTANT allows them to form a hierarchy, so that one class may be a finer division of another. ASSISTANT does not form a decision tree iteratively in the manner of ID3, but does include algorithms for choosing a 'good' training set from the objects available. ASSISTANT has been used in several medical domains with promising results.

The bottom-most three systems in the figure are commercial derivatives of ACLS. While they do not significantly advance the underlying theory, they incorporate many user-friendly innovations and utilities that expedite the task of generating and using decision trees. They all have industrial successes to their credit. Westinghouse Electric's Water Reactor Division, for example, points to a fuel-enrichment application in which the company was able to boost revenue by 'more than ten million dollars per annum' through the use of one of them.[1]

3. The induction task

We now give a more precise statement of the induction task. The basis is a universe of *objects* that are described in terms of a collection of *attributes*. Each attribute measures some important feature of an object and will be limited here to taking a (usually small) set of discrete, mutually exclusive values. For example, if the objects were Saturday mornings and the classification task involved the weather, attributes might be

outlook, with values {sunny, overcast, rain}
temperature, with values {cool, mild, hot}
humidity, with values {high, normal}
windy, with values {true, false}

Taken together, the attributes provide a zeroth-order language for characterizing objects in the universe. A particular Saturday morning might be described as

outlook: overcast
temperature: cool
humidity: normal
windy: false

[1] Letter cited in the journal *Expert Systems* (January, 1985), p. 20.

Each object in the universe belongs to one of a set of mutually exclusive *classes*. To simplify the following treatment, we will assume that there are only two such classes denoted *P* and *N*, although the extension to any number of classes is not difficult. In two-class induction tasks, objects of class P and N are sometimes referred to as *positive instances* and *negative instances*, respectively, of the concept being learned.

The other major ingredient is a *training set* of objects whose class is known. The induction task is to develop a *classification rule* that can determine the class of any object from its values of the attributes. The immediate question is whether or not the attributes provide sufficient information to do this. In particular, if the training set contains two objects that have identical values for each attribute and yet belong to different classes, it is clearly impossible to differentiate between these objects with reference only to the given attributes. In such a case attributes will be termed *inadequate* for the training set and hence for the induction task.

As mentioned above, a classification rule will be expressed as a decision tree. Table 1 shows a small training set that uses the 'Saturday morning' attributes. Each object's value of each attribute is shown, together with the class of the object (here, class P mornings are suitable for some unspecified activity). A decision tree that correctly classifies each object in the training set is given in Figure 2. Leaves of a decision tree are class names, other nodes represent attribute-based tests with a branch for each possible outcome. In order to classify an object, we start at the root of the tree, evaluate the test, and take the branch appropriate to the outcome. The process continues until a leaf is encountered, at which time the object is asserted to belong to

Table 1. A small training set

No.	Attributes				Class
	Outlook	Temperature	Humidity	Windy	
1	sunny	hot	high	false	N
2	sunny	hot	high	true	N
3	overcast	hot	high	false	P
4	rain	mild	high	false	P
5	rain	cool	normal	false	P
6	rain	cool	normal	true	N
7	overcast	cool	normal	true	P
8	sunny	mild	high	false	N
9	sunny	cool	normal	false	P
10	rain	mild	normal	false	P
11	sunny	mild	normal	true	P
12	overcast	mild	high	true	P
13	overcast	hot	normal	false	P
14	rain	mild	high	true	N

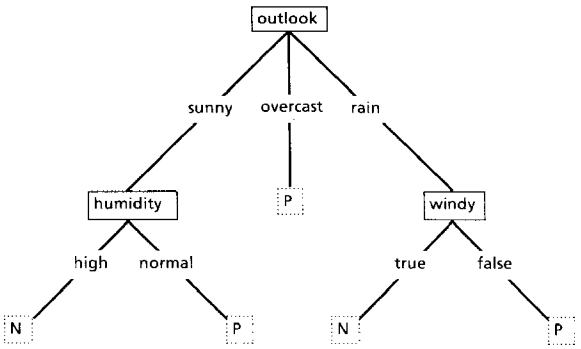


Figure 2. A simple decision tree

the class named by the leaf. Taking the decision tree of Figure 2, this process concludes that the object which appeared as an example at the start of this section, and which is not a member of the training set, should belong to class P. Notice that only a subset of the attributes may be encountered on a particular path from the root of the decision tree to a leaf; in this case, only the outlook attribute is tested before determining the class.

If the attributes are adequate, it is always possible to construct a decision tree that correctly classifies each object in the training set, and usually there are many such correct decision trees. The essence of induction is to move beyond the training set, i.e. to construct a decision tree that correctly classifies not only objects from the training set but other (unseen) objects as well. In order to do this, the decision tree must capture some meaningful relationship between an object's class and its values of the attributes. Given a choice between two decision trees, each of which is correct over the training set, it seems sensible to prefer the simpler one on the grounds that it is more likely to capture structure inherent in the problem. The simpler tree would therefore be expected to classify correctly more objects outside the training set. The decision tree of Figure 3, for instance, is also correct for the training set of Table 1, but its greater complexity makes it suspect as an 'explanation' of the training set.[2]

4. ID3

One approach to the induction task above would be to generate all possible decision trees that correctly classify the training set and to select the simplest of them. The

[2] The preference for simpler trees, presented here as a commonsense application of Occam's Razor, is also supported by analysis. Pearl (1978b) and Quinlan (1983a) have derived upper bounds on the expected error using different formalisms for generalizing from a set of known cases. For a training set of predetermined size, these bounds increase with the complexity of the induced generalization.

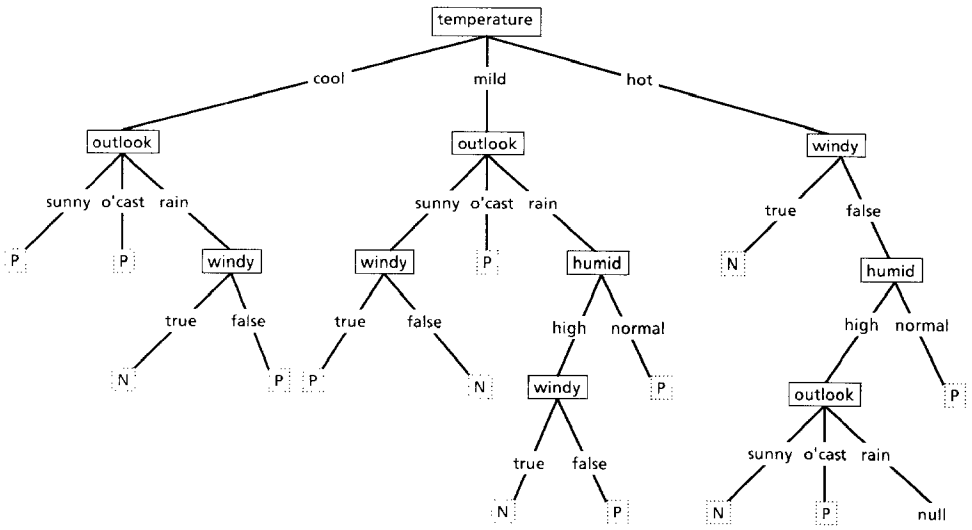


Figure 3. A complex decision tree.

number of such trees is finite but very large, so this approach would only be feasible for small induction tasks. ID3 was designed for the other end of the spectrum, where there are many attributes and the training set contains many objects, but where a reasonably good decision tree is required without much computation. It has generally been found to construct simple decision trees, but the approach it uses cannot guarantee that better trees have not been overlooked.

The basic structure of ID3 is iterative. A subset of the training set called the *window* is chosen at random and a decision tree formed from it; this tree correctly classifies all objects in the window. All other objects in the training set are then classified using the tree. If the tree gives the correct answer for all these objects then it is correct for the entire training set and the process terminates. If not, a selection of the incorrectly classified objects is added to the window and the process continues. In this way, correct decision trees have been found after only a few iterations for training sets of up to thirty thousand objects described in terms of up to 50 attributes. Empirical evidence suggests that a correct decision tree is usually found more quickly by this iterative method than by forming a tree directly from the entire training set. However, O'Keefe (1983) has noted that the iterative framework cannot be guaranteed to converge on a final tree unless the window can grow to include the entire training set. This potential limitation has not yet arisen in practice.

The crux of the problem is how to form a decision tree for an arbitrary collection C of objects. If C is empty or contains only objects of one class, the simplest decision tree is just a leaf labelled with the class. Otherwise, let T be any test on an object with possible outcomes $O_1, O_2, \ldots O_w$. Each object in C will give one of these outcomes for T, so T produces a partition $\{C_1, C_2, \ldots C_w\}$ of C with C_i containing those ob-

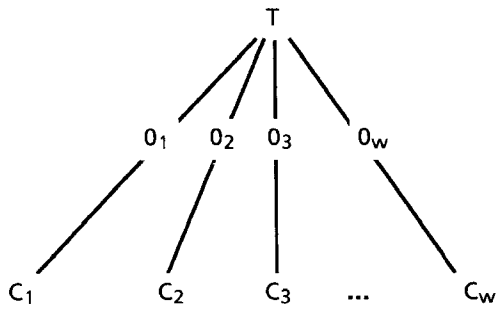


Figure 4. A tree structuring of the objects in C.

jects having outcome O_i. This is represented graphically by the tree form of Figure 4. If each subset C_i in this figure could be replaced by a decision tree for C_i, the result would be a decision tree for all of C. Moreover, so long as two or more C_i's are non-empty, each C_i is smaller than C. In the worst case, this divide-and-conquer strategy will yield single-object subsets that satisfy the one-class requirement for a leaf. Thus, provided that a test can always be found that gives a non-trivial partition of any set of objects, this procedure will always produce a decision tree that correctly classifies each object in C.

The choice of test is crucial if the decision tree is to be simple. For the moment, a test will be restricted to branching on the values of an attribute, so choosing a test comes down to selecting an attribute for the root of the tree. The first induction programs in the ID series used a seat-of-the-pants evaluation function that worked reasonably well. Following a suggestion of Peter Gacs, ID3 adopted an information-based method that depends on two assumptions. Let C contain p objects of class P and n of class N. The assumptions are:

(1) Any correct decision tree for C will classify objects in the same proportion as their representation in C. An arbitrary object will be determined to belong to class P with probability $p/(p+n)$ and to class N with probability $n/(p+n)$.
(2) When a decision tree is used to classify an object, it returns a class. A decision tree can thus be regarded as a source of a message 'P' or 'N', with the expected information needed to generate this message given by

$$I(p, n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

If attribute A with values $\{A_1, A_2, \ldots A_v\}$ is used for the root of the decision tree, it will partition C into $\{C_1, C_2, \ldots C_v\}$ where C_i contains those objects in C that have value A_i of A. Let C_i contain p_i objects of class P and n_i of class N. The expected

information required for the subtree for C_i is $I(p_i, n_i)$. The expected information required for the tree with A as root is then obtained as the weighted average

$$E(A) = \sum_{i=1}^{v} \frac{p_i + n_i}{p + n} I(p_i, n_i)$$

where the weight for the ith branch is the proportion of the objects in C that belong to C_i. The information gained by branching on A is therefore

$$gain(A) = I(p, n) - E(A)$$

A good rule of thumb would seem to be to choose that attribute to branch on which gains the most information.[3] ID3 examines all candidate attributes and chooses A to maximize gain(A), forms the tree as above, and then uses the same process recursively to form decision trees for the residual subsets $C_1, C_2, \ldots C_v$.

To illustrate the idea, let C be the set of objects in Table 1. Of the 14 objects, 9 are of class P and 5 are of class N, so the information required for classification is

$$I(p, n) = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.940 \text{ bits}$$

Now consider the outlook attribute with values {sunny, overcast, rain}. Five of the 14 objects in C have the first value (sunny), two of them from class P and three from class N, so

$$p_1 = 2 \quad n_1 = 3 \quad I(p_1, n_1) = 0.971$$

and similarly

$$p_2 = 4 \quad n_2 = 0 \quad I(p_2, n_2) = 0$$
$$p_3 = 3 \quad n_3 = 2 \quad I(p_3, n_3) = 0.971$$

The expected information requirement after testing this attribute is therefore

$$E(\text{outlook}) = \frac{5}{14} I(p_1, n_1) + \frac{4}{14} I(p_2, n_2) + \frac{5}{14} I(p_3, n_3)$$
$$= 0.694 \text{ bits}$$

[3] Since I(p,n) is constant for all attributes, maximizing the gain is equivalent to minimizing E(A), which is the mutual information of the attribute A and the class. Pearl (1978a) contains an excellent account of the rationale of information-based heuristics.

The gain of this attribute is then

$$\text{gain(outlook)} = 0.940 - \text{E(outlook)} = 0.246 \text{ bits}$$

Similar analysis gives

$$\text{gain(temperature)} = 0.029 \text{ bits}$$
$$\text{gain(humidity)} = 0.151 \text{ bits}$$
$$\text{gain(windy)} = 0.048 \text{ bits}$$

so the tree-forming method used in ID3 would choose outlook as the attribute for the root of the decision tree. The objects would then be divided into subsets according to their values of the outlook attribute and a decision tree for each subset would be induced in a similar fashion. In fact, Figure 2 shows the actual decision tree generated by ID3 from this training set.

A special case arises if C contains no objects with some particular value A_j of A, giving an empty C_j. ID3 labels such a leaf as 'null' so that it fails to classify any object arriving at that leaf. A better solution would generalize from the set C from which C_j came, and assign this leaf the more frequent class in C.

The worth of ID3's attribute-selecting heuristic can be assessed by the simplicity of the resulting decision trees, or, more to the point, by how well those trees express real relationships between class and attributes as demonstrated by the accuracy with which they classify objects other than those in the training set (their *predictive accuracy*). A straightforward method of assessing this predictive accuracy is to use only part of the given set of objects as a training set, and to check the resulting decision tree on the remainder.

Several experiments of this kind have been carried out. In one domain, 1.4 million chess positions described in terms of 49 binary-valued attributes gave rise to 715 distinct objects divided 65%:35% between the classes. This domain is relatively complex since a correct decision tree for all 715 objects contains about 150 nodes. When training sets containing 20% of these 715 objects were chosen at random, they produced decision trees that correctly classified over 84% of the unseen objects. In another version of the same domain, 39 attributes gave 551 distinct objects with a correct decision tree of similar size; training sets of 20% of these 551 objects gave decision trees of almost identical accuracy. In a simpler domain (1,987 objects with a correct decision tree of 48 nodes), randomly-selected training sets containing 20% of the objects gave decision trees that correctly classified 98% of the unseen objects. In all three cases, it is clear that the decision trees reflect useful (as opposed to random) relationships present in the data.

This discussion of ID3 is rounded off by looking at the computational requirements of the procedure. At each non-leaf node of the decision tree, the gain of each untested attribute A must be determined. This gain in turn depends on the values p_i

and n_i for each value A_i of A, so every object in C must be examined to determine its class and its value of A. Consequently, the computational complexity of the procedure at each such node is $O(|C| \cdot |A|)$, where $|A|$ is the number of attributes above. ID3's total computational requirement per iteration is thus proportional to the product of the size of the training set, the number of attributes and the number of non-leaf nodes in the decision tree. The same relationship appears to extend to the entire induction process, even when several iterations are performed. No exponential growth in time or space has been observed as the dimensions of the induction task increase, so the technique can be applied to large tasks.

5. Noise

So far, the information supplied in the training set has been assumed to be entirely accurate. Sadly, induction tasks based on real-world data are unlikely to find this assumption to be tenable. The description of objects may include attributes based on measurements or subjective judgements, both of which may give rise to errors in the values of attributes. Some of the objects in the training set may even have been misclassified. To illustrate the idea, consider the task of developing a classification rule for medical diagnosis from a collection of patient histories. An attribute might test for the presence of some substance in the blood and will almost inevitably give false positive or negative readings some of the time. Another attribute might assess the patient's build as slight, medium, or heavy, and different assessors may apply different criteria. Finally, the collection of case histories will probably include some patients for whom an incorrect diagnosis was made, with consequent errors in the class information provided in the training set.

What problems might errors of these kinds pose for the tree-building procedure described earlier? Consider again the small training set in Table 1, and suppose now that attribute outlook of object 1 is incorrectly recorded as overcast. Objects 1 and 3 will then have identical descriptions but belong to different classes, so the attributes become inadequate for this training set. The attributes will also become inadequate if attribute windy of object 4 is corrupted to true, because that object will then conflict with object 14. Finally, the initial training set can be accounted for by the simple decision tree of Figure 2 containing 8 nodes. Suppose that the class of object 3 were corrupted to N. A correct decision tree for this corrupted training set would now have to explain the apparent special case of object 3. The smallest such tree contains twelve nodes, half again as complex as the 'real' tree. These illustrations highlight two problems: errors in the training set may cause the attributes to become inadequate, or may lead to decision trees of spurious complexity.

Non-systematic errors of this kind in either the values of attributes or class information are usually referred to as *noise*. Two modifications are required if the tree-building algorithm is to be able to operate with a noise-affected training set.

(1) The algorithm must be able to work with inadequate attributes, because noise can cause even the most comprehensive set of attributes to appear inadequate.
(2) The algorithm must be able to decide that testing further attributes will not improve the predictive accuracy of the decision tree. In the last example above, it should refrain from increasing the complexity of the decision tree to accommodate a single noise-generated special case.

We start with the second requirement of deciding when an attribute is really relevant to classification. Let C be a collection of objects containing representatives of both classes, and let A be an attribute with random values that produces subsets $\{C_1, C_2, \ldots C_v\}$. Unless the proportion of class P objects in each of the C_i is exactly the same as the proportion of class P objects in C itself, branching on attribute A will give an apparent information gain. It will therefore appear that testing attribute A is a sensible step, even though the values of A are random and so cannot help to classify the objects in C.

One solution to this dilemma might be to require that the information gain of any tested attribute exceeds some absolute or percentage threshold. Experiments with this approach suggest that a threshold large enough to screen out irrelevant attributes also excludes attributes that are relevant, and the performance of the tree-building procedure is degraded in the noise-free case.

An alternative method based on the chi-square test for stochastic independence has been found to be more useful. In the previous notation, suppose attribute A produces subsets $\{C_1, C_2, \ldots C_v\}$ of C, where C_i contains p_i and n_i objects of class P and N, respectively. If the value of A is irrelevant to the class of an object in C, the expected value p'_i of p_i should be

$$p'_i = p \cdot \frac{p_i + n_i}{p + n}$$

If n'_i is the corresponding expected value of n_i, the statistic

$$\sum_{i=1}^{v} \frac{(p_i - p'_i)^2}{p'_i} + \frac{(n_i - n'_i)^2}{n'_i}$$

is approximately chi-square with v-1 degrees of freedom. Provided that none of the values p'_i or n'_i are very small, this statistic can be used to determine the confidence with which one can reject the hypothesis that A is independent of the class of objects in C (Hogg and Craig, 1970). The tree-building procedure can then be modified to prevent testing any attribute whose irrelevance cannot be rejected with a very high (e.g. 99%) confidence level. This has been found effective in preventing over-

complex trees that attempt to 'fit the noise' without affecting performance of the procedure in the noise-free case.[4]

Turning now to the first requirement, we see that the following situation can arise: a collection of C objects may contain representatives of both classes, yet further testing of C may be ruled out, either because the attributes are inadequate and unable to distinguish among the objects in C, or because each attribute has been judged to be irrelevant to the class of objects in C. In this situation it is necessary to produce a leaf labelled with class information, but the objects in C are not all of the same class.

Two possibilities suggest themselves. The notion of class could be generalized to allow the value $p/(p+n)$ in the interval (0,1), a class of 0.8 (say) being interpreted as 'belonging to class P with probability 0.8'. An alternative approach would be to opt for the more numerous class, i.e. to assign the leaf to class P if $p > n$, to class N if $p < n$, and to either if $p = n$. The first approach minimizes the sum of the squares of the error over objects in C, while the second minimizes the sum of the absolute errors over objects in C. If the aim is to minimize expected error, the second approach might be anticipated to be superior, and indeed this has been found to be the case.

Several studies have been carried out to see how this modified procedure holds up under varying levels of noise (Quinlan 1983b, 1985a). One such study is outlined here based on the earlier-mentioned task with 551 objects and 39 binary-valued attributes. In each experiment, the whole set of objects was artificially corrupted as described below and used as a training set to produce a decision tree. Each object was then corrupted anew, classified by this tree and the error rate determined. This process was repeated twenty times to give more reliable averages.

In this study, values were corrupted as follows. A noise level of n percent applied to a value meant that, with probability n percent, the true value was replaced by a value chosen at random from among the values that could have appeared.[5] Table 2 shows the results when noise levels varying from 5% to 100% were applied to the values of the most noise-sensitive attribute, to the values of all attributes simultaneously, and to the class information. This table demonstrates the quite different forms of degradation observed. Destroying class information produces a linear increase in error so that, when all class information is noise, the resulting decision tree classifies objects entirely randomly. Noise in a single attribute does not have a dramatic effect. Noise in all attributes together, however, leads to a relatively rapid increase in error which reaches a peak and declines. The peak is somewhat inter-

[4] ASSISTANT uses an information-based measure to perform much the same function, but no comparative results are available to date.

[5] It might seem that the value should be replaced by an incorrect value. Consider, however, the case of a two-valued attribute corrupted with 100% noise. If the value of each object were replaced by the (only) incorrect value, the initial attribute will have been merely inverted with no loss of information.

Table 2. Error rates produced by noise in a single attribute, all attributes, and class information

Noise level	Single attribute	All attributes	Class information
5%	1.3%	11.9%	2.6%
10%	2.5%	18.9%	5.5%
15%	3.3%	24.6%	8.3%
20%	4.6%	27.8%	9.9%
30%	6.1%	29.5%	14.8%
40%	7.6%	30.3%	18.1%
50%	8.8%	29.2%	21.8%
60%	9.4%	27.5%	26.4%
70%	9.9%	25.9%	27.2%
80%	10.4%	26.0%	29.5%
90%	10.8%	25.6%	34.1%
100%	10.8%	25.9%	49.6%

esting, and can be explained as follows. Let C be a collection of objects containing p from class P and n from class N, respectively. At noise levels around 50%, the algorithm for constructing decision trees may still find relevant attributes to branch on, even though the performance of this tree on unseen but equally noisy objects will be essentially random. Suppose the tree for C classifies objects as class P with probability $p/(n+p)$. The expected error if objects with a similar class distribution to those in C were classified by this tree is given by

$$\frac{p}{p+n} \cdot \left(1 - \frac{p}{p+n}\right) + \frac{n}{p+n} \cdot \left(1 - \frac{n}{p+n}\right) = \frac{2pn}{(p+n)^2}$$

At very high levels of noise, however, the algorithm will find all attributes irrelevant and classify everything as the more frequent class; assume without loss of generality that this class is P. The expected error in this case is

$$\frac{p}{p+n} \cdot 0 + \frac{n}{p+n} \cdot 1 = \frac{n}{p+n}$$

which is less than the above expression since we have assumed that p is greater than n. The decline in error is thus a consequence of the chi-square cutoff coming into play as noise becomes more intense.

The table brings out the point that low levels of noise do not cause the tree-building machinery to fall over a cliff. For this task, a 5% noise level in a single attribute produces a degradation in performance of less than 2%; a 5% noise level in all attributes together produces a 12% degradation in classification performance; while a similar

noise level in class information results in a 3% degradation. Comparable figures have been obtained for other induction tasks.

One interesting point emerged from other experiments in which a correct decision tree formed from an uncorrupted training set was used to classify objects whose descriptions were corrupted. This scenario corresponds to forming a classification rule under controlled and sanitized laboratory conditions, then using it to classify objects in the field. For higher noise levels, the performance of the correct decision tree on corrupted data was found to be inferior to that of an imperfect decision tree formed from data corrupted to a similar level! (This phenomenon has an explanation similar to that given above for the peak in Table 2.) The moral seems to be that it is counter-productive to eliminate noise from the attribute information in the training set if these same attributes will be subject to high noise levels when the induced decision tree is put to use.

6. Unknown attribute values

The previous section examined modifications to the tree-building process that enabled it to deal with noisy or corrupted values. This section is concerned with an allied problem that also arises in practice: unknown attribute values. To continue the previous medical diagnosis example, what should be done when the patient case histories that are to form the training set are incomplete?

One way around the problem attempts to fill in an unknown value by utilizing information provided by context. Using the previous notation, let us suppose that a collection C of objects contains one whose value of attribute A is unknown. ASSISTANT (Kononenko *et al*, 1984) uses a Bayesian formalism to determine the probability that the object has value A_i of A by examining the distribution of values of A in C as a function of their class. Suppose that the object in question belongs to class P. The probability that the object has value A_i for attribute A can be expressed as

$$\text{prob}(A = A_i \mid \text{class} = P) = \frac{\text{prob}(A = A_i \;\&\; \text{class} = P)}{\text{prob}(\text{class} = P)} = \frac{p_i}{p}$$

where the calculation of p_i and p is restricted to those members of C whose value of A is known. Having determined the probability distribution of the unknown value over the possible values of A, this method could either choose the most likely value or divide the object into fractional objects, each with one possible value of A, weighted according to the probabilities above.

Alen Shapiro (private communication) has suggested using a decision-tree approach to determine the unknown values of an attribute. Let C′ be the subset of C consisting of those objects whose value of attribute A is defined. In C′, the original

Table 3. Proportion of times that an unknown attribute value is replaced by an incorrect value

Replacement method	Attribute		
	1	2	3
Bayesian	28%	27%	38%
Decision tree	19%	22%	19%
Most common value	28%	27%	40%

class (P or N) is regarded as another attribute while the value of attribute A becomes the 'class' to be determined. That is, C′ is used to construct a decision tree for determining the value of attribute A from the other attributes and the class. When constructed, this decision tree can be used to 'classify' each object in C − C′ and the result assigned as the unknown value of A.

Although these methods for determining unknown attribute values look good on paper, they give unconvincing results even when only a single value of one attribute is unknown; as might be expected, their performance is much worse when several values of several attributes are unknown. Consider again the 551-object 39-attribute task. We may ask how well the methods perform when asked to fill in a single unknown attribute value. Table 3 shows, for each of the three most important attributes, the proportion of times each method fails to replace an unknown value by its correct value. For comparison, the table also shows the same figure for the simple strategy: always replace an unknown value of an attribute with its most common value. The Bayesian method gives results that are scarcely better than those given by the simple strategy and, while the decision-tree method uses more context and is thereby more accurate, it still gives disappointing results.

Rather than trying to guess unknown attribute values, we could treat 'unknown' as a new possible value for each attribute and deal with it in the same way as other values. This can lead to an anomalous situation, as shown by the following example. Suppose A is an attribute with values $\{A_1, A_2\}$ and let C be a collection of objects such that

$$p_1 = 2 \quad p_2 = 2$$
$$n_1 = 2 \quad n_2 = 2$$

giving a value of 1 bit for E(A). Now let A′ be an identical attribute except that one of the objects with value A_1 of A has an unknown value of A′. A′ has the values $\{A'_1, A'_2, A'_3 = \text{unknown}\}$, so the corresponding values might be

$$p'_1 = 1 \quad p'_2 = 2 \quad p'_3 = 1$$
$$n'_1 = 2 \quad n'_2 = 2 \quad n'_3 = 0$$

resulting in a value of 0.84 bits for E(A′). In terms of the selection criterion developed earlier, A′ now seems to give a higher information gain than A. Thus, having unknown values may apparently increase the desirability of an attribute, a result entirely opposed to common sense. The conclusion is that treating 'unknown' as a separate value is not a solution to the problem.

One strategy which has been found to work well is as follows. Let A be an attribute with values $\{A_1, A_2, \ldots A_v\}$. For some collection C of objects, let the numbers of objects with value A_i of A be p_i and n_i, and let p_u and n_u denote the numbers of objects of class P and N respectively that have unknown values of A. When the information gain of attribute A is assessed, these objects with unknown values are distributed across the values of A in proportion to the relative frequency of these values in C. Thus the gain is assessed as if the true value of p_i were given by

$$p_i + p_u \cdot \text{ratio}_i$$

where

$$\text{ratio}_i = \frac{p_i + n_i}{\sum_i (p_i + n_i)}$$

and similarly for n_i. (This expression has the property that unknown values can only decrease the information gain of an attribute.) When an attribute has been chosen by the selection criterion, objects with unknown values of that attribute are discarded before forming decision trees for the subsets $\{C_i\}$.

The other half of the story is how unknown attribute values are dealt with during classification. Suppose that an object is being classified using a decision tree that wishes to branch on attribute A, but the object's value of attribute A is unknown. The correct procedure would take the branch corresponding to the real value A_i but, since this value is unknown, the only alternative is to explore all branches without forgetting that some are more probable than others.

Conceptually, suppose that, along with the object to be classified, we have been passed a *token* with some value T. In the situation above, each branch of A_i is then explored in turn, using a token of value

$$T \cdot \text{ratio}_i$$

i.e. the given token value is distributed across all possible values in proportion to the ratios above. The value passed to a branch may be distributed further by subsequent tests on other attributes for which this object has unknown values. Instead of a single path to a leaf, there may now be many, each qualified by its token value. These token values at the leaves are summed for each class, the result of the classification being

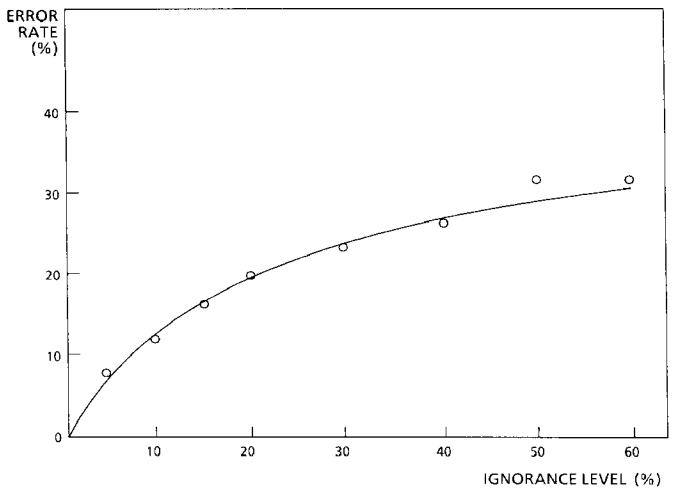


Figure 5. Error produced by unknown attribute values.

that class with the higher value. The distribution of values over the possible classes might also be used to compute a confidence level for the classification.

Straightforward though it may be, this procedure has been found to give a very graceful degradation as the incidence of unknown values increases. Figure 5 summarizes the results of an experiment on the now-familiar task with 551 objects and 39 attributes. Various 'ignorance levels' analogous to the earlier noise levels were explored, with twenty repititions at each level. For each run at an ignorance level of m percent, a copy of the 551 objects was made, replacing each value of every attribute by 'unknown' with m percent probability. A decision tree for these (incomplete) objects was formed as above, and then used to classify a new copy of each object corrupted in the same way. The figure shows that the degradation of performance with ignorance level is gradual. In practice, of course, an ignorance level even as high as 10% is unlikely – this would correspond to an average of one value in every ten of the object's description being unknown. Even so, the decision tree produced from such a patchy training set correctly classifies nearly ninety percent of objects that also have unknown values. A much lower level of degradation is observed when an object with unknown values is classified using a correct decision tree.

This treatment has assumed that no information whatsoever is available regarding an unknown attribute. Catlett (1985) has taken this approach a stage further by allowing partial knowledge of an attribute value to be stated in Shafer notation (Garvey, Lowrance and Fischler, 1981). This notation permits probabilistic assertions to be made about any subset or subsets of the possible values of an attribute that an object might have.

7. The selection criterion

Attention has recently been refocussed on the evaluation function for selecting the best attribute-based test to form the root of a decision tree. Recall that the criterion described earlier chooses the attribute that gains most information. In the course of their experiments, Bratko's group encountered a medical induction problem in which the attribute selected by the gain criterion ('age of patient', with nine value ranges) was judged by specialists to be less relevant than other attributes. This situation was also noted on other tasks, prompting Kononenko *et al* (1984) to suggest that the gain criterion tends to favor attributes with many values.

Analysis supports this finding. Let A be an attribute with values A_1, A_2, ... A_v and let A' be an attribute formed from A by splitting one of the values into two. If the values of A were sufficiently fine for the induction task at hand, we would not expect this refinement to increase the usefulness of A. Rather, it might be anticipated that excessive fineness would tend to obscure structure in the training set so that A' was in fact less useful than A. However, it can be proved that gain(A') is greater than or equal to gain(A), being equal to it only when the proportions of the classes are the same for both subdivisions of the original value. In general, then, gain(A') will exceed gain(A) with the result that the evaluation function of Section 4 will prefer A' to A. By analogy, attributes with more values will tend to be preferred to attributes with fewer.

As another way of looking at the problem, let A be an attribute with random values and suppose that the set of possible values of A is large enough to make it unlikely that two objects in the training set have the same value for A. Such an attribute would have maximum information gain, so the gain criterion would select it as the root of the decision tree. This would be a singularly poor choice since the value of A, being random, contains no information pertinent to the class of objects in the training set.

ASSISTANT (Kononenko *et al*, 1984) solves this problem by requiring that all tests have only two outcomes. If we have an attribute A as before with v values A_1, A_2, ... A_v, the decision tree no longer has a branch for each possible value. Instead, a subset S of the values is chosen and the tree has two branches, one for all values in the set and one for the remainder. The information gain is then computed as if all values in S were amalgamated into one single attribute value and all remaining values into another. Using this selection criterion (the *subset* criterion), the test chosen for the root of the decision tree uses the attribute and subset of its values that maximizes the information gain. Kononenko *et al* report that this modification led to smaller decision trees with an improved classification performance. However, the trees were judged to be less intelligible to human beings, in agreement with a similar finding of Shepherd (1983).

Limiting decision trees to a binary format harks back to CLS, in which each test was of the form 'attribute A has value A_i', with two branches corresponding to true and false. This is clearly a special case of the test implemented in ASSISTANT, which

permits a set of values, rather than a single value, to be distinguished from the others.

It is also worth noting that the method of dealing with attributes having continuous values follows the same binary approach. Let A be such an attribute and suppose that the distinct values of A that occur in C are sorted to give the sequence $V_1, V_2, \ldots, V_k$. Each pair of values V_i, V_{i+1} suggests a possible threshold

$$\frac{V_i + V_{i+1}}{2}$$

that divides the objects of C into two subsets, those with a value of A above and below the threshold respectively. The information gain of this division can then be investigated as above.

If all tests must be binary, there can be no bias in favor of attributes with large numbers of values. It could be argued, however, that ASSISTANT's remedy has undesirable side-effects that have to be taken into account. First, it can lead to decision trees that are even more unintelligible to human experts than is ordinarily the case, with unrelated attribute values being grouped together and with multiple tests on the same attribute.

More importantly, the subset criterion can require a large increase in computation. An attribute A with v values has 2^v value subsets and, when trivial and symmetric subsets are removed, there are still $2^{v-1}-1$ different ways of specifying the distinguished subset of attribute values. The information gain realized with each of these must be investigated, so a single attribute with v values has a computational requirement similar to $2^{v-1}-1$ binary attributes. This is not of particular consequence if v is small, but the approach would appear infeasible for an attribute with 20 values.

Another method of overcoming the bias is as follows. Consider again our training set containing p and n objects of class P and N respectively. As before, let attribute A have values $A_1, A_2, \ldots A_v$ and let the numbers of objects with value A_i of attribute A be p_i and n_i respectively. Enquiring about the value of attribute A itself gives rise to information, which can be expressed as

$$IV(A) = -\sum_{i=1}^{v} \frac{p_i+n_i}{p+n} \log_2 \frac{p_i+n_i}{p+n}$$

IV(A) thus measures the information content of the answer to the question, 'What is the value of attribute A?' As discussed earlier, gain(A) measures the reduction in the information requirement for a classification rule if the decision tree uses attribute A as a root. Ideally, as much as possible of the information provided by determining the value of an attribute should be useful for classification purposes or, equivalently, as little as possible should be 'wasted'. A good choice of attribute would then be one for which the ratio

$$\text{gain(A)} \;/\; \text{IV(A)}$$

is as large as possible. This ratio, however, may not always be defined – IV(A) may be zero – or it may tend to favor attributes for which IV(A) is very small. The *gain ratio* criterion selects, from among those attributes with an average-or-better gain, the attribute that maximizes the above ratio.

This can be illustrated by returning to the example based on the training set of Table 1. The information gain of the four attributes is given in Section 4 as

$$\begin{aligned} \text{gain(outlook)} &= 0.246 \text{ bits} \\ \text{gain(temperature)} &= 0.029 \text{ bits} \\ \text{gain(humidity)} &= 0.151 \text{ bits} \\ \text{gain(windy)} &= 0.048 \text{ bits} \end{aligned}$$

Of these, only outlook and humidity have above-average gain. For the outlook attribute, five objects in the training set have the value sunny, four have overcast and five have rain. The information obtained by determining the value of the outlook attribute is therefore

$$\begin{aligned} \text{IV(outlook)} &= -\frac{5}{14}\log_2\frac{5}{14} - \frac{4}{14}\log_2\frac{4}{14} - \frac{5}{14}\log_2\frac{5}{14} \\ &= 1.578 \text{ bits} \end{aligned}$$

Similarly,

$$\text{IV(humidity)} = -\frac{7}{14}\log_2\frac{7}{14} - \frac{7}{14}\log_2\frac{7}{14} = 1 \text{ bit}$$

So,

$$\begin{aligned} \text{gain ratio(outlook)} &= 0.246 \;/\; 1.578 = 0.156 \\ \text{gain ratio(humidity)} &= 0.151 \;/\; 1.000 = 0.151 \end{aligned}$$

The gain ratio criterion would therefore still select the outlook attribute for the root of the decision tree, although its superiority over the humidity attribute is now much reduced.

The various selection criteria have been compared empirically in a series of experiments (Quinlan, 1985b). When all attributes are binary, the gain ratio criterion has been found to give considerably smaller decision trees: for the 551-object task, it produces a tree of 143 nodes compared to the smallest previously-known tree of 175 nodes. When the task includes attributes with large numbers of values, the subset criterion gives smaller decision trees that also have better predictive performance, but can require much more computation. However, when these many-valued attributes are augmented by redundant attributes which contain the same information at a

lower level of detail, the gain ratio criterion gives decision trees with the greatest predictive accuracy. All in all, these experiments suggest that the gain ratio criterion does pick a good attribute for the root of the tree. Testing an attribute with many values, however, will fragment the training set C into very small subsets $\{C_i\}$ and the decision trees for these subsets may then have poor predictive accuracy. In such cases, some mechanism such as value subsets or redundant attributes is needed to prevent excessive fragmentation.

The three criteria discussed here are all information-based, but there is no reason to suspect that this is the only possible basis for such criteria. Recall that the modifications to deal with noise barred an attribute from being used in the decision tree unless it could be shown to be relevant to the class of objects in the training set. For any attribute A, the value of the statistic presented in Section 5, together with the number v of possible values of A, determines the confidence with which we can reject the null hypothesis that an object's value of A is irrelevant to its class. Hart (1985) has proposed that this same test could function directly as a selection criterion: simply pick the attribute for which this confidence level is highest. This measure takes explicit account of the number of values of an attribute and so may not exhibit bias. Hart notes, however, that the chi-square test is valid only when the expected values of p'_i and n'_i are uniformly larger than four. This condition could be violated by a set C of objects either when C is small or when few objects in C have a particular value of some attribute, and it is not clear how such sets would be handled. No empirical results with this approach are yet available.

8. Conclusion

The aim of this paper has been to demonstrate that the technology for building decision trees from examples is fairly robust. Current commercial systems are powerful tools that have achieved noteworthy successes. The groundwork has been done for advances that will permit such tools to deal even with noisy, incomplete data typical of advanced real-world applications. Work is continuing at several centers to improve the performance of the underlying algorithms.

Two examples of contemporary research give some pointers to the directions in which the field is moving. While decision trees generated by the above systems are fast to execute and can be very accurate, they leave much to be desired as representations of knowledge. Experts who are shown such trees for classification tasks in their own domain often can identify little familiar material. It is this lack of familiarity (and perhaps an underlying lack of modularity) that is the chief obstacle to the use of induction for building large expert systems. Recent work by Shapiro (1983) offers a possible solution to this problem. In his approach, called *Structured Induction*, a rule-formation task is tackled in the same style as structured programming. The task is solved in terms of a collection of notional super-attributes, after which the subtasks

of inducing classification rules to find the values of the super-attributes are approached in the same top-down fashion. In one classification problem studied, this method reduced a totally opaque, large decision tree to a hierarchy of nine small decision trees, each of which 'made sense' to an expert.

ID3 allows only two classes for any induction task, although this restriction has been removed in most later systems. Consider, however, the task of developing a rule from a given set of examples for classifying an animal as a monkey, giraffe, elephant, horse, etc. A single decision tree could be produced in which these various classes appeared as leaves. An alternative approach taken by systems such as INDUCE (Michalski, 1980) would produce a collection of classification rules, one to discriminate monkeys from non-monkeys, another to discriminate giraffes from non-giraffes, and so on. Which approach is better? In a private communication, Marcel Shoppers has set out an argument showing that the latter can be expected to give more accurate classification of objects that were not in the training set. The multi-tree approach has some associated problems – the separate decision trees may classify an animal as both a monkey and a giraffe, or fail to classify it as anything, for example – but if these can be sorted out, this approach may lead to techniques for building more reliable decision trees.

Acknowledgements

It is a pleasure to acknowledge the stimulus and suggestions provided over many years by Donald Michie, who continues to play a central role in the development of this methodology. Ivan Bratko, Igor Kononenko, Igor Mosetic and other members of Bratko's group have also been responsible for many insights and constructive criticisms. I have benefited from numerous discussions with Ryszard Michalski, Alen Shapiro, Jason Catlett and other colleagues. I am particularly grateful to Pat Langley for his careful reading of the paper in draft form and for the many improvements he recommended.

References

Buchanan, B.G., & Mitchell, T.M. (1978). Model-directed learning of production rules. In D.A. Waterman, F. Hayes-Roth (Eds.), *Pattern directed inference systems*. Academic Press.

Carbonell, J.G., Michalski, R.S., & Mitchell, T.M. (1983). An overview of machine learning, In R.S. Michalski, J.G. Carbonell and T.M. Mitchell, (Eds.), *Machine learning: An artificial intelligence approach*. Palo Alto: Tioga Publishing Company.

Catlett, J. (1985). *Induction using the shafer representation* (Technical report). Basser Department of Computer Science, University of Sydney, Australia.

Dechter, R., & Michie, D. (1985). *Structured induction of plans and programs* (Technical report). IBM Scientific Center, Los Angeles, CA.

Feigenbaum, E.A., & Simon, H.A. (1963). Performance of a reading task by an elementary perceiving and memorizing program, *Behavioral Science, 8.*

Feigenbaum, E.A. (1981). Expert systems in the 1980s. In A. Bond (Ed.), *State of the art report on machine intelligence.* Maidenhead: Pergamon-Infotech.

Garvey, T.D., Lowrance, J.D., & Fischler, M.A. (1981). An inference technique for integrating knowledge from disparate sources. *Proceedings of the Seventh International Joint Conference on Artificial Intelligence.* Vancouver, B.C., Canada: Morgan Kaufmann.

Hart, A.E. (1985). Experience in the use of an inductive system in knowledge engineering. In M.A. Bramer (Ed.), *Research and development in expert systems.* Cambridge University Press.

Hogg, R.V., & Craig, A.T. (1970). *Introduction to mathematical statistics.* London: Collier-Macmillan.

Hunt, E.B. (1962). *Concept learning: An information processing problem.* New York: Wiley.

Hunt, E.B., Marin, J., & Stone, P.J. (1966). *Experiments in induction.* New York: Academic Press.

Kononenko, I., Bratko, I., & Roskar, E. (1984). *Experiments in automatic learning of medical diagnostic rules* (Technical report). Jozef Stefan Institute, Ljubljana, Yugoslavia.

Langley, P., Bradshaw, G.L., & Simon, H.A. (1983). Rediscovering chemistry with the BACON system. In R.S. Michalski, J.G. Carbonell and T.M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach.* Palo Alto: Tioga Publishing Company.

Michalski, R.S (1980). Pattern recognition as rule-guided inductive inference. *IEEE Transactions on Pattern Analysis and Machine Intelligence 2.*

Michalski, R.S., & Stepp, R.E. (1983). Learning from observation: conceptual clustering. In R.S. Michalski, J.G. Carbonell & T.M. Mitchell (Eds.), *Machine learning: An artificial intelligence approach.* Palo Alto: Tioga Publishing Company.

Michie, D. (1982). Experiments on the mechanisation of game-learning 2 – Rule-based learning and the human window. *Computer Journal 25.*

Michie, D. (1983). Inductive rule generation in the context of the Fifth Generation. *Proceedings of the Second International Machine Learning Workshop.* University of Illinois at Urbana-Champaign.

Michie, D. (1985). Current developments in Artificial Intelligence and Expert Systems. In *International Handbook of Information Technology and Automated Office Systems.* Elsevier.

Nilsson, N.J. (1965). *Learning machines,* New York: McGraw-Hill.

O'Keefe, R.A. (1983). Concept formation from very large training sets. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence.* Karlsruhe, West Germany: Morgan Kaufmann.

Patterson, A., & Niblett, T. (1983). *ACLS user manual.* Glasgow: Intelligent Terminals Ltd.

Pearl, J. (1978a). Entropy, information and rational decisions (Technical report). Cognitive Systems Laboratory, University of California, Los Angeles.

Pearl, J. (1978b). On the connection between the complexity and credibility of inferred models. *International Journal of General Systems, 4.*

Quinlan, J.R. (1969). A task-independent experience gathering scheme for a problem solver. *Proceedings of the First International Joint Conference on Artificial Intelligence.* Washington, D.C.: Morgan Kaufmann.

Quinlan, J.R. (1979). Discovering rules by induction from large collections of examples. In D. Michie (Ed.), *Expert systems in the micro electronic age.* Edinburgh University Press.

Quinlan, J.R. (1982). Semi-autonomous acquisition of pattern-based knowledge. In J.E. Hayes, D. Michie & Y-H. Pao (Eds.), *Machine intelligence 10.* Chichester: Ellis Horwood.

Quinlan, J.R. (1983a). Learning efficient classification procedures and their application to chess endgames. In R.S. Michalski, J.G. Carbonell & T.M. Mitchell, (Eds.), *Machine learning: An artificial intelligence approach.* Palo Alto: Tioga Publishing Company.

Quinlan, J.R. (1983b). Learning from noisy data, *Proceedings of the Second International Machine Learning Workshop.* University of Illinois at Urbana-Champaign.

Quinlan, J.R. (1985a). The effect of noise on concept learning. In R.S. Michalski, J.G. Carbonell & T.M.

Mitchell (Eds.), *Machine learning*. Los Altos: Morgan Kaufmann (in press).

Quinlan, J.R. (1985b). Decision trees and multi-valued attributes. In J.E. Hayes & D. Michie (Eds.), *Machine intelligence 11*. Oxford University Press (in press).

Sammut, C.A. (1985). Concept development for expert system knowledge bases. *Australian Computer Journal 17*.

Samuel, A. (1967). Some studies in machine learning using the game of checkers II: Recent progress. *IBM J. Research and Development 11*.

Shapiro, A. (1983). *The role of structured induction in expert systems*. Ph.D. Thesis, University of Edinburgh.

Shepherd, B.A. (1983). An appraisal of a decision-tree approach to image classification. *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*. Karlsruhe, West Germany: Morgan Kaufmann.

Winston, P.H. (1975). Learning structural descriptions from examples. In P.H. Winston (Ed.), *The psychology of computer vision*. McGraw-Hill.

Available online at www.sciencedirect.com
ScienceDirect
Decision Support Systems 42 (2006) 674–689
Decision Support Systems
www.elsevier.com/locate/dsw

A new approach to classification based on association rule mining

Guoqing Chen *, Hongyan Liu, Lan Yu, Qiang Wei, Xing Zhang

Department of Management Science and Engineering, School of Economics and Management, Tsinghua University, Beijing 100084, China

Received 19 February 2004; received in revised form 9 March 2005; accepted 9 March 2005
Available online 25 July 2005

Abstract

Classification is one of the key issues in the fields of decision sciences and knowledge discovery. This paper presents a new approach for constructing a classifier, based on an extended association rule mining technique in the context of classification. The characteristic of this approach is threefold: first, applying the information gain measure to the generation of candidate itemsets; second, integrating the process of frequent itemsets generation with the process of rule generation; third, incorporating strategies for avoiding rule redundancy and conflicts into the mining process. The corresponding mining algorithm proposed, namely GARC (Gain based Association Rule Classification), produces a classifier with satisfactory classification accuracy, compared with other classifiers (e.g., C4.5, CBA, SVM, NN). Moreover, in terms of association rule based classification, GARC could filter out many candidate itemsets in the generation process, resulting in a much smaller set of rules than that of CBA.

Keywords: Data mining; Association rule; Classification; Information gain

1. Introduction

Classification is one of the key issues in the field of decision sciences, a field which plays an important role in supporting business and scientific decision-making. In recent years, it has also been one of the focal points in data mining and knowledge discovery. Classification is finding a classifier that results from training datasets with predetermined targets, fine-tuning it with test datasets, and using it to classify other datasets of interest. There exists various ways of constructing classifiers in the form of, for example, rules, decision trees, Bayesian networks, support vectors machine, etc. [12,14–16,21,24,26,29–31]. Decision trees classifiers, such as Quinlan's C4.5/5.0 classifier and its extensions [30], have received considerable attention due to its speed and understandability. Moreover, a number of efforts have been put forward to focus on the various aspects of improvements [5,9,25,33]. Another type of classification technique that has attracted an increasing number of attempts in recent years is finding classification rules based on association rule mining techniques, e.g., Refs. [4,20–23,29].

A classification rule is of the form $X \Rightarrow C$, where X is a set of data items, and C is a class (label) and a predetermined target. With such a rule, a transaction or data record t in a given database could be classified into class C if t contains X. Apparently, a classifica-

* Corresponding author. Tel.: +86 10 62772940; fax: +86 10 62785876.
E-mail address: chengq@em.tsinghua.edu.cn (G. Chen).

0167-9236/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.dss.2005.03.005

tion rule could be regarded as an association rule of a special kind.

Roughly speaking, an association rule is a relationship between data items. Two measures, namely the Degree of Support (D_{supp}) and the Degree of Confidence (D_{conf}), are used to define a rule. For example, a rule like "Milk⇒Diaper with D_{supp}=20%, D_{conf}=80%" means that "20% of the customers bought both Milk and Diaper" and that "80% of the customers who bought Milk also bought Diaper". That is, D_{supp} corresponds to statistical significance, while D_{conf} is a measure of the rule's strength [3].

Formally, let $\boldsymbol{I}=\{I_i,\ i=1,\ldots,s\}$ be a set of items. A transaction database T is a set of transactions, where each transaction t is a set of items such that $t \subseteq \boldsymbol{I}$. An association rule is of the form $X \Rightarrow Y$, where $X \subset \boldsymbol{I}$, $Y \subset \boldsymbol{I}$ are called itemsets, and $X \cap Y = \varnothing$. A transaction t is called to contain X, if $X \subseteq t$. Let $D_{\text{supp}}(X)$ be the fraction of transactions that contain X in a database T, $D_{\text{supp}}(X) = \|X\|/|T|$. The degree of support and degree of confidence for a rule $X \Rightarrow Y$ are defined as follows:

$$D_{\text{supp}}(X \Rightarrow Y) = \|X \cup Y\|/|T|$$

$$D_{\text{conf}}(X \Rightarrow Y) = \|X \cup Y\|/\|X\|$$

where X and Y are itemsets with $X \cap Y = \varnothing$, T is the set of all the transactions contained in the database concerned, $\|X\|$ is the number of the transactions in T that contain X, $\|X \cup Y\|$ is the number of the transactions in T that contain X and Y, and $|T|$ is the number of the transactions in T. In other words, $D_{\text{supp}}(X \Rightarrow Y)$ is the percentage of transactions containing both X and Y in the whole dataset, while $D_{\text{conf}}(X \Rightarrow Y)$ is the ratio of the number of transactions that contain X and Y over the number of transactions that contain X. They are used to evaluate a rule against given thresholds, minimal support α and minimal confidence β, respectively. In particular, if D_{supp} of an itemset X is no less than α (i.e., $D_{\text{supp}}(X) \geq \alpha$), then X is called a frequent itemset, otherwise called an excluded itemset. There have been many efforts proposed to discover association rules in various ways [1,2,8,11–13,17,28,32,34,37], among which the Apriori algorithm by Agrawal and Srikant [1] is usually deemed as a classical algorithm.

In the classification based on association rules mining, a well-known method, namely the CBA method proposed by Liu et al. [21] and its modifications [20,23], uses an Apriori-type association rule mining approach [1] to generate classification rules, which usually generates all the frequent itemsets, followed by the rule generation process. Subsequently, filters may be applied to the rules so as to eliminate non-interesting ones such as conflicts and so on. In other words, basically, CBA directly employs the Apriori-type approach for a particular kind of association rule, namely classification rules in forms of $X \Rightarrow C$. Thus, its efficiency heavily relies on the process of generating frequent itemsets. Like conventional association rules, classification rules are generated based on all the frequent itemsets generated. Then these rules are sorted according to a filtering measure, if desired.

While classifiers in forms of rules are often appealing for use and explanation by decision makers, directly applying the Apriori-type approach may however result in a large number of itemsets and then of rules, which would further increase the effort for understanding the rules as well as for resolving rule redundancy and conflicts. Therefore, it is considered desirable if some strategies such as itemset reduction and redundancy/conflict resolutions could be incorporated into the process of frequent itemsets generation, such that fewer itemsets need to be generated and therefore with fewer resultant rules. Apparently, a smaller set of classification rules is often preferable than a larger set at the same level of accuracy in terms of rule understandability.

Moreover, in the process of frequent itemsets generation, the Apriori-type method usually considers all the combinations of items in candidate itemsets. With massive datasets, the number of these combinations is generally very large. In fact, different items in these combinations may play different roles in measuring the degrees of support and degrees of confidence. Therefore, it is deemed desirable if only a part of the items (e.g., those "informative" ones) in candidate itemsets need to be considered in generating frequent itemsets.

This paper addresses some of the above-mentioned issues and presents a new approach for constructing a classifier, based on an extended association rule mining technique in the context of classification. Section 2 describes the issues of concern along with the notion of information gain to be used in the mining process to reduce the number of

candidate itemsets, as well as with the notions and certain related properties of rule redundancy and conflicts. In Section 3, the new mining algorithm called GARC (Gain based Association Rule Classification) is presented, which combines the processes of frequent itemsets generation and rule generation, where the measures for redundancy and conflicts avoidance and information gain are incorporated. Finally, results and respective analyses of data experiments are provided in Section 4.

2. Classification rules

2.1. Basic notions

As mentioned previously, the classification rules mining problem can be regarded as a special case of the association rules mining problem [21,22,27]. The task of classification is to find a set of rules so as to identify the classes of undetermined transactions. In classification, a classifier is usually built based upon a dataset that is divided into two groups: one is for training, and the other for testing, each consisting of data items and class labels. In terms of association rules, these class labels are special cases of items. For the sake of clarity, we hereafter refer to them separately, otherwise indicated where necessary.

Let T be the dataset with each transaction composed of a number of distinctive items in the set of all items $\boldsymbol{I}$ and a class label in $G=\{C_1, C_2, \ldots, C_g\}$, X be a subset of $\boldsymbol{I}$ (i.e., $X \subseteq \boldsymbol{I}$), and C_k be a class label in $G(k=1, 2,\ldots,g)$. Notably, in classification-oriented association rule mining, only those rules each with one single class label as its consequent need to be considered; therefore in this paper, each itemset (such as XC_k) is used to represent a rule (such as $X \Rightarrow C_k$) identically. In other words, itemset XC_k corresponds to rule $X \Rightarrow C_k$, with $D_{\text{supp}}(XC_k)=||XC_k|| / |T|$, and $D_{\text{conf}}(XC_k)=||XC_k|| / ||X||$. A transaction t in T is called to contain X if $t \supseteq X$. XC_k is called to be a p-itemset, if X contains p items. If $D_{\text{supp}}(XC_k) \geq \alpha$, then XC_k is called a frequent itemset. Furthermore, if $D_{\text{conf}}(XC_k) \geq \beta$, then XC_k is called a qualified itemset, and can be used to produce a classification rule such as $X \Rightarrow C_k$. For a rule $X \Rightarrow C_k$, sometimes X is referred to as the antecedent of the rule and C_k as the consequent of the rule. Moreover, for the sake of convenience, two parameters namely *lcount* and *wcount* are sometimes used to denote $||X||$ and $||XC_k||$, as the number of transactions containing X and the number of transactions containing XC_k, respectively. Thus, mining classification rules is used to discover such qualified association rules as $X \Rightarrow C_k$, for $k=1, 2, \ldots, g$.

2.2. Information gain

Information gain is one of the measures used to select best split attributes in decision tree classifiers [7,35]. In this paper, it could also be used as a measure to reduce the number of itemsets. In the process of frequent itemsets generation, instead of considering all the combinations of items in candidate itemsets in the Apriori-type method, information gain measure will be used to select the best attribute. In this way, only those items containing the best item with maximum information gain need to be selected to further generate candidate itemsets.

Suppose an attribute A has n distinct values that partition the training dataset T into subsets T_1, T_2, ..., T_n. For a dataset $S \subseteq T$, freq(C_k, S) represents the number of transactions in S that belong to class C_k. Then info*(S)* is defined as follows to measure the average amount of information needed to identify the class of a transaction in S:

$$\text{info}(S) = -\sum_{k=1}^{g} \frac{\text{freq}(C_k, S)}{|S|} \times \log_2\left(\frac{\text{freq}(C_k, S)}{|S|}\right)$$

where $|S|$ is the number of transactions in S and g is the number of classes.

After the dataset T is partitioned in accordance with n values of attribute A, the expected information requirement could be defined as:

$$\text{info}_A(T) = \sum_{i=1}^{n} \frac{|T_i|}{|T|} \times \text{info}(T_i)$$

The information gained by partitioning T according to attribute A is defined as:

$$\text{gain}(A) = \text{info}(T) - \text{info}_A(T)$$

Among all attributes in dataset T, the best split attribute is the one that maximizes the information gain.

2.3. Rule redundancy and conflicts

Though classification rules can be discovered using Apriori-type association rule mining techniques directly, the whole set of classification rules (i.e., rules satisfying α and β) might be poor in quality. First, the number of classification rules may be too large to easily construct classifiers. More seriously, from the viewpoint of classification, there may exist conflicting rules (e.g., $X \Rightarrow C_i$ and $X \Rightarrow C_j$) and redundant rules (e.g., $X \Rightarrow C_i$ and $XY \Rightarrow C_i$, with $D_{\text{conf}}(X \Rightarrow C_i) \geq D_{\text{conf}}(XY \Rightarrow C_i)$). The conflicting rules will lead to identifying a transaction into two classes, while the redundancy will result in a rule like $XY \Rightarrow C_i$ that is semantically meaningless for classification (given $X \Rightarrow C_i$).

Definition 2.1. Rule r is called to precede rule r' if either $D_{\text{conf}}(r) > D_{\text{conf}}(r')$, or $D_{\text{conf}}(r) = D_{\text{conf}}(r')$ and $D_{\text{supp}}(r) > D_{\text{supp}}(r')$.

Definition 2.2. Let Ψ be a set of discovered classification rules. Then rule $Z \Rightarrow C_i$ in Ψ is called redundant if there already exists a rule $X \Rightarrow C_i$ in Ψ such that $Z \supset X$. Moreover, for $i \neq j$, rules $Z \Rightarrow C_j$ and $X \Rightarrow C_i$ in Ψ are called conflicting if there already exists a rule $X \Rightarrow C_i$ in Ψ such that either $Z = X$, or $Z \supset X$ and $Z \Rightarrow C_j$ does not precede $X \Rightarrow C_i$.

Apparently, coping with such rule redundancy and conflicts is desirable because otherwise (1) a transaction containing X may be classified into two classes (e.g., C_i and C_j), (2) a rule (e.g., $XY \Rightarrow C_i$) may not be regarded useful (i.e., redundant) for identifying a transaction due to the existence of another rule (e.g., $X \Rightarrow C_i$), and (3) a transaction containing XY may be classified into two classes (e.g., C_i and C_j); or $XY \Rightarrow C_j$ is not significant enough to be used, compared with $X \Rightarrow C_i$. It is worth mentioning, however, that these notions of redundancy and conflict are particularly relevant for classification and may not be of concern for association rules in general.

Usually, given a (nonempty) set Ψ of discovered classification rules, i.e., $\Psi = \{r | r$ is a classification rule, $D_{\text{supp}}(r) \geq \alpha$ and $D_{\text{conf}}(r) \geq \beta\}$, filters can be built to deal with the redundancy and conflicts. It can be seen that for any nonempty Ψ there exists a corresponding nonempty set Ψ_c of rules with such redundancy and conflicts removed. Ψ_c is referred to as a compact set of Ψ. A constructive way to obtain Ψ_c is a repetitive resolution procedure as follows:

First set $\Psi_c = \Psi$, then repeat the following steps until no further changes are made for Ψ_c.

(i) check each rule $XY \Rightarrow C_i$ in Ψ_c, if there exists a rule $X \Rightarrow C_i$ in Ψ_c, then delete $XY \Rightarrow C_i$. That is, $\Psi_c = \Psi_c - \{XY \Rightarrow C_i\}$.
(ii) check each rule $X \Rightarrow C_i$ in Ψ_c, if there exists a rule $X \Rightarrow C_j$ in Ψ_c that does not precede $X \Rightarrow C_i$, then delete $X \Rightarrow C_j$. That is, $\Psi_c = \Psi_c - \{X \Rightarrow C_j\}$.
(iii) check each rule $XY \Rightarrow C_j$ in Ψ_c, if there exists a rule $X \Rightarrow C_i$ in Ψ_c that precedes it, then delete $XY \Rightarrow C_j$. That is, $\Psi_c = \Psi_c - \{XY \Rightarrow C_j\}$.

Obviously, Ψ_c is nonempty if Ψ is nonempty. Moreover, Ψ_c is not unique, depending on the order in which the above three steps are performed. Notably, in this paper, our primary attention is not paid to developing a separate filter to derive Ψ_c from Ψ, but to exploring certain ways to avoid rule conflicts and redundancy, which could be incorporated in the integrated mining process.

2.4. Strategies in avoiding rule conflicts and redundancy

When the strategies are incorporated into the process of generating Ψ, the rules in Ψ will be free of redundancy and conflict. More importantly, these strategies will help identify excluded (i.e., not frequent) itemsets inside the process of candidate itemsets generation, resulting in fewer itemsets to be generated.

With regard to the redundancy stated in Definition 2.2, the strategy that could be applied is that, if $X \Rightarrow C_i$ holds, then any candidate itemset containing XC_i need not to be produced, because any such itemsets as XYC_i (i.e., $Z = XY$) would not be regarded semantically necessary from the perspective of classification. In other words, if $X \Rightarrow C_i$ holds, it means that any transaction containing X will be classified into class C_i, including any transaction containing XY. Note that the number of candidate itemsets to generate is reduced.

Next, consider rule conflicts for $Z = X$ in Definition 2.2. The following theorem indicates that it can simply be avoided if the pre-specified minimal confidence β is set to be over 0.5, which is regarded reasonable in many real applications.

Theorem 2.1. *If $\beta > 50\%$, then rules $X \Rightarrow C_i$ and $X \Rightarrow C_j$ will not hold in T simultaneously.*

Proof. Without loss of generality, assume that $X \Rightarrow C_i$ holds in T with $D_{\text{conf}}(X \Rightarrow C_i) \geq \beta > 50\%$. For $g \geq 2$, since $\|X\| = \|XC_i\| + \sum_{\substack{j=1 \\ j \neq i}}^{g} \|XC_j\|$, and $\frac{\|X\|}{\|X\|} = \frac{\|XC_i\|}{\|X\|} + \sum_{\substack{j=1 \\ j \neq i}}^{g} \frac{\|XC_j\|}{\|X\|}$, then

$$\sum_{\substack{j=1 \\ j \neq i}}^{g} \frac{\|XC_j\|}{\|X\|} = 1 - D_{\text{conf}}(X \Rightarrow C_i).$$

From $\frac{\|XC_j\|}{\|X\|} \geq 0$, for j=1, 2,…,$g$, we have $\frac{\|XC_j\|}{\|X\|} \leq 1 - Dconf(X \rightarrow C_i) < 50\% < \beta$, which means that $X \Rightarrow C_j$ does not hold in T. □

In other words, if both $X \Rightarrow C_i$ and $X \Rightarrow C_j$ hold simultaneously, both of them must involve two mutually disjoint sets of transactions (denoted as T_{XC_i} and T_{XC_j}) containing XC_i and XC_j respectively. Otherwise, if a transaction t is involved in generating both rules, we will have $XC_iC_j \subseteq t$, which is a contradiction to the structure of t, for t contains only a single class label of G={C_1,C_2,…,C_g}. That is, $T_{XC_i} \cap T_{XC_j} = \varnothing$. Since these two sets are the subsets of T_X (where T_X is the set of transactions containing X), we have $|T_{XC_i}|/|T_X| + |T_{XC_j}|/|T_X| \leq |T_X|/|T_X| = 1$. Semantically, the following relationship exists: $\|X_{C_i}\|/\|X\| + \|X_{C_j}\|/\|X\| \leq \|X\|/\|X\| = 1$, which means that $D_{\text{conf}}(X \Rightarrow C_i) + D_{\text{conf}}(X \Rightarrow C_j) \leq 1$. Apparently, however, this is a contradiction to the supposition that $D_{\text{conf}}(X \Rightarrow C_j) \leq \beta > 0.5$ and $D_{\text{conf}}(X \Rightarrow C_i) \geq \beta > 0.5$. In brief, the strategy to set β to be over 0.5 will prevent the rule conflict from happening.

In addition, the following theorem can be used to further reduce the number of candidate itemsets. This also corresponds to rule conflicts stated in Definition 2.2. It will be proved in Theorem 2.2 that, if $X \Rightarrow C_i$ holds in T and if $D_{\text{conf}}(X \Rightarrow C_i) > 1 - \alpha$ or $D_{\text{supp}}(X) < 2\alpha$, then $XY \Rightarrow C_j$ does not hold in T.

Theorem 2.2. *Suppose rule $X \Rightarrow C_i$ holds in T, (1) if $1 - D_{conf}(X \Rightarrow C_i) < \alpha$, then any itemset like XYC_j is an excluded itemset; (2) if $||X|| < 2|T|\alpha$, then any itemset like XYC_j is an excluded itemset; where $Y \cap X = \varnothing$, $Y \cap C_k = \varnothing$, $k = 1, 2, \ldots, g$, and $g \geq 2$.*

Proof. (1) Since $\|X\| = \|XC_i\| + \sum_{\substack{j=1 \\ j \neq i}}^{g} \|XC_j\|$, then $\frac{\|XC_j\|}{\|X\|} \leq 1 - D_{\text{conf}}(X \Rightarrow C_i)$. □

Further, $\frac{\|XYC_j\|}{\|X\|} \leq \frac{\|XC_j\|}{\|X\|} \leq 1 - D_{\text{conf}}(X \Rightarrow C_i)$, *thus* $\frac{\|XYC_j\|}{|T|} \leq \frac{\|XYC_j\|}{\|X\|} \leq 1 - D_{\text{conf}}(X \Rightarrow C_i)$.

Then since $1 - D_{\text{conf}}(X \Rightarrow Ci) < \alpha$, then $\frac{\|XYC_j\|}{|T|} < \alpha$, which means $D_{\text{supp}}(XYC_j) < \alpha$. That is, XYC_j is an excluded itemset.

(2) Since $\|X\| < 2|T|\alpha$, then $\frac{\|X\| - \|XC_i\|}{|T|} < \frac{2|T|\alpha - \|XC_i\|}{|T|}$.

From $\frac{\|XC_i\|}{|T|} \geq \alpha$, we have $\|XC_i\| \geq |T|\alpha$.

Then $\frac{\|X\| - \|XC_i\|}{|T|} < \frac{2|T|\alpha - |T|\alpha}{|T|} = \alpha$.

Since $\frac{\|XYC_j\|}{|T|} \leq \frac{\|X\| - \|XC_i\|}{|T|}$, $D\text{supp}(XYC_j) < \alpha$.

That is, XYC_j is an excluded itemset.

Thus, if rule $X \Rightarrow C_i$ holds in T, and the conditions of Theorem 2.2 are satisfied, then the rule conflicts can be avoided, because hereby any itemset like XYC_j is an excluded itemset. Accordingly, this strategy may be applied to the mining process, in which the itemset, XYC_j, does not need to be considered further in generating larger candidate itemsets.

Example 1. Given a dataset as shown in Table 1, with $\beta = 0.8$ and $\alpha = 0.21$. After the first scan of the dataset, rule "overcast⇒play (D_{conf}=1, D_{supp}=0.29)" can be obtained. Before executing the second scan, it has been already known that any larger candidate itemset such as {overcast, *Y*, don't play} is an excluded itemset, because $1 - 1 < \alpha$ according to Theorem 2.2, where $Y \subseteq$ {Temperature, Humidity, Windy}.

Table 1
Training dataset

TID	Outlook	Temperature	Humidity	Windy	Class
1	Sunny	Mild	Normal	True	Play
2	Sunny	Hot	High	True	Don't play
3	Sunny	Hot	High	False	Don't play
4	Sunny	Mild	High	False	Don't play
5	Sunny	Cool	Normal	False	Play
6	Overcast	Mild	High	True	Play
7	Overcast	Hot	High	False	Play
8	Overcast	Cool	Normal	True	Play
9	Overcast	Hot	Normal	False	Play
10	Rain	Mild	High	True	Don't play
11	Rain	Cool	Normal	True	Don't play
12	Rain	Mild	High	False	Play
13	Rain	Cool	High	False	Play
14	Rain	Mild	High	False	Play
15	Overcast	Cool	High	False	Don't play

If another transaction as follows is added to Table 1:

15	Overcast	Cool	high	False	Don't Play

then D_{conf} (overcast⇒play)=0.8, so we have $1-D_{\text{conf}}$ (overcast⇒play)=1−0.8=0.2<0.21. Likewise, {overcast, *Y*, don't play} is an excluded itemset. Suppose that the class label of transaction 9 in Table 1 is changed to be *Don't play*, with $\beta=0.7$ and $\alpha=0.15$. Then from $\|\text{overcast}\|=4<2|T|\alpha=2\times 14\times 0.15=4.2$ (Theorem 2.2), itemsets {overcast, play} and {overcast, don't play} can be excluded from the itemsets used to generate larger candidate itemsets.

3. Discovering classification rules

In this section, an algorithm called GARC (Gain based Association Rule Classification) will be presented, which could discover the compact set of classification rules. Though the general idea is in the spirit of association rule mining, it differs from conventional CBA techniques that directly apply the Apriori-type association rule mining procedures. The main characteristic of the proposed algorithm is threefold. First, it combines the conventional itemset generation and rule generation processes, and makes use of the information maintained for both rule itemsets and excluded itemsets. Second, the information gain measure is incorporated so as to only generate the itemsets including the best-split attribute value, which leads to a reduction of candidate itemsets. Third, certain strategies are applied into the mining process such that conflicting/redundant rules are avoided as well as the number of candidate itemsets generated is reduced. As a result, the resultant compact set is more condensed and understandable (in terms of fewer rules), and in the mean time, as revealed by data experiments in the next sections, the classification accuracy turns out to be satisfactory.

3.1. GARC: gain based association rule classification

Generally speaking, one transaction in *T* with *s* items can generate around 2^s candidate itemsets. To cope with this, the information gain measure is first used here to reduce the number of candidate itemsets. That is, only those candidate itemsets including the best split attribute value will be generated. Concretely, after the first scan of the database, all of 1-itemsets can be obtained and saved in a variable named *Cand*. According to the *lcount* and *wcount* values of each candidate itemset, information gain for each attribute *A*, which could be used to partition the database *T* into *n* datasets, may be calculated as follows:

$$\begin{aligned}\text{info}_A(T) &= \sum_{i=1}^{n}\frac{|T_i|}{|T|}\times\text{info}(T_i) = -\sum_{i=1}^{n}\frac{|T_i|}{|T|}\\ &\quad\times\left(\sum_{k=1}^{g}\frac{\text{freq}(C_j,T_i)}{|T_i|}\times\log_2\frac{\text{freq}(C_j,T_i)}{|T_i|}\right)\\ &= -\sum_{i=1}^{n}D_{\text{supp}}(A=v_i)\\ &\quad\times\left(\sum_{k=1}^{g}D\text{c}_{\text{conf}}(i_k)\times\log_2 D_{\text{conf}}(i_k)\right)\end{aligned}$$

where T_i corresponds to the dataset whose attribute *A*'s value equals v_i, *g* is the number of classes, i_k represents itemset $\{v_i, C_k\}$. As a result, a best split attribute (called *bestattr*) can be selected after the first scan of the database. Then during the next scan of the database, only those itemsets containing this best split attribute specified by *bestattr* will be generated. The following example helps illustrate the idea.

Example 2. Let us consider Table 1 again. After the first scan of the dataset in Table 1, among the four attributes, attribute *outlook* is selected as the best split attribute. Then during the second scan of the database, tuple 1 can produce the following three candidate itemsets: {sunny, mild, play}, {sunny, normal, play}, and {sunny, true, play}. Note that {mild, normal, play}, {mild, true, play}, and {normal, true, play} will not be generated.

In addition to information gain, certain conflicts/redundancy avoidance strategies are used to improve the quality of the rule set as well as to reduce the number of candidate itemsets, which is detailed in the next subsection, along with how excluded itemsets are dealt with.

3.2. Algorithmic details

As stated previously, an itemset *XC* is interchangeably referred to as a rule $X\Rightarrow C$. A qualified itemset

Table 2
Algorithm GARC

```
Algorithm GARC:
1.  rule={r|r is an 1-itemset, D_supp(r)≥α and D_conf(r)≥β};
    //initiating the set of qualified itemsets//
2.  excluded={e|e is an 1-itemset, and D_supp(e)<α}; //initiating
    the set of excluded itemsets//
3.  bestattr=gain;
4.  if (β≤0.5) and (∀r:X⇒C_i ∈ rule, ∃r′:X⇒C_j ∈ rule
    such that r′ does not precede r) then
5.     rule=rule−{X⇒C_j};          //deleting conflicting rules//
6.  for k from 2 to m do //m is the number of ancetedent
    attributes//
7.     empty(cand); // emptying cand, the set of candidate
       itemsets//
8.     if coverall(rule)
9.        break;
10.    for each transaction t in T do
11.       C_t=CandidateGen(t, bestattr, k);
12.       for each c ∈ C_t do
13.          maintCand(rule, excluded, c, cand)
14.       end for;
15.    end for;
16.    R={r|r ∈ cand, D_supp(r)≥α and D_conf(r)≥β};
       //the set of qualified k-itemsets//
17.    if (β≤0.5) and (∀r:X⇒C ∈ R, ∃r′:X⇒C_j ∈ R
       such that r′ does not precede r) then
18.       R=R−{X⇒C_j};             //deleting conflicting rules//
19.    if (∀r:X⇒C_i ∈ R, ∃r′:XY⇒C_j ∈ R such that Y≠∅,
       X∩Y=∅, and r′ does not precede r) then
20.       R=R−{XY⇒C_j};            //deleting conflicting rules//
21.    rule=rule∪R;
22.    E={e|e ∈ cand, D_supp(e)<α};
23.    excluded=excluded∪E;
24. end for;
25. sort(rule);
```

XC corresponds to a qualified rule $X \Rightarrow C$. In addition, for XC, X is referred to the antecedent of XC, denoted by antecedent(XC)=X. The main algorithm is shown in Table 2.

Lines 1–3 perform the first scan of the database. It produces all the 1-itemsets from which qualified 1-itemsets are generated. By the method described above, the best split attribute is selected by *gain*, which employs the information gain measure and helps reduce the number of candidate itemsets (Table 3). Lines 6–24 perform the consecutive scans of the database. *coverall(rule)* tests whether rules already contain all of transactions in the training dataset, and if true, the main iteration breaks. During each scan, for a certain transaction t, *CandidateGen(t, bestattr, k)* generates all k-itemsets with each containing the *bestattr*. Working on these itemsets in Ct, *maintCand(rule, excluded, c, cand)* then generates and maintains candidate itemsets according to qualified itemsets and excluded itemsets. Note that the itemsets returned by *maintCand* will be redundancy-free, and will not produce any conflicting rules if the conditions of Theorem 2.2 are satisfied. In the mean time, this will lead to generating fewer itemsets inside the process. Moreover, since rule conflicts with regard to Theorem 2.1 will be avoided if its condition ($\beta > 0.5$) is satisfied, lines 4–5 and 17–20 further remove conflicting rules (r') when the conditions of Theorems 2.1 and 2.2 do not hold. The advantage of incorporating the conflict resolution strategy at the stages inside the k-itemset generation process (rather than after the process as a separate filter) is to further reduce the number of candidate itemsets generated (for $k \geq 1$). Finally, all rules are sorted, when the main procedure is terminated.

Note that each of the rules included in the classifier built by the above algorithm satisfies the pre-specified minimal support and minimal confidence thresholds (i.e., (α and β). Moreover, the rules that cannot be predicated to be excluded itemsets according to Theorem 2.2 (line 6 in Table 4) will be added to the set of candidate itemsets and further counted. If c contains a qualified itemset or an excluded itemset, the class attribute in c is substituted by a fixed mark q that is different from all class labels of G (i.e., G is the set of all class labels) in the database for the purpose of counting *lcount* while not affecting *wcount* (Tables 4 and 5). In addition,

Table 3
Sub-algorithm *gain*

```
gain
begin
  for each attribute A_i ∈{A_1, A_2... A_m} do
    compute info(A_i) using D_supp and D_conf values of all 1-itemsets
  end for
  bestattr=A_1; mininfo=info(A_1);
  for each attribute A_i ∈ {A_1, A_2... A_m} do
    if mininfo>info(A_i) then
      mininfo=info(A_i)
      bestattr=A_i
    end if
  end for
  return bestattr
end
```

Table 4
Sub-algorithm *maintCand*

```
maintCand(rule, excluded, c, cand)
1. begin
2.   if¬∃r ∈ rule, (c⊃r) and¬∃e ∈ excluded, (c⊃e) then
3.     if¬∃r ∈ rule, (c⊃ancetedent(r)) then
4.       addToCand(c, cand); //adding c into cand//
5.     else
6.       if ∃r∈ rule, (c⊃ancetedent(r)) and
         ((1−D_conf(r)≥α) and (D_supp(X)≥2α) then
7.         addToCand(c, cand);
8.       else //when c is an excluded itemset//
9.         excluded=excluded∪E;
10.      end if;
11.    end if;
12.  else
13.    if ∃e ∈ excluded, (c⊃e) or ∃r∈ rule, (c⊃r) then
14.      c′=ancetedent(e)∪{q}; //q is a fixed mark different
         from any class in G//
15.      addToCand(c′, cand); //adding c′ into cand//
16.    end if;
17.  end if;
18. end;
```

the rules that are redundant according to Definition 2.2 will not be included (generated) in *cand* (line 13 in Table 4, and line 14 in Table 5).

Finally, the algorithm will terminate in a finite number of *m* passes at most, where *m* is the number of attributes. Notably, the set of resultant classification rules is a compact set. Moreover, if the discovered rule set without using the redundancy/conflict resolution strategies is not empty (e.g., if the set of qualified 1-itemsets is not empty), the compact set will not be empty either.

4. Experimental results

This section shows an empirical performance evaluation of algorithm GARC, along with some comparisons with other algorithms. The experiments consist of five parts. The first part is to compare GARC with C4.5-type [30], CBA [21], NN [10], and SVM [37] classifiers on accuracy. The second part of the experiments is to test how the pruning strategies affect the efficiency and further examines the execution time of GARC. The third part discusses the impact of minimal support and minimal confidence thresholds on GARC outcomes. In the fourth part, the use of information gain for rule reduction is examined. The last part compares GARC with the CBA classifier in terms of the number of rules produced. The experiments were conducted in the environment with Windows 2000 Server, Intel Pentium 4 1.5 GHz, 512 MB RAM and Visual C++. It should be mentioned that, all the following experiments are tested based on datasets from a commonly used benchmarking database in the field, namely the UCI Machine Learning Repository [27], including the 26 datasets that CBA method selected. In total, 30 datasets are used.

The basic information of the datasets is listed in Table 6.

Since some data are continuous and Apriori-type methods mainly focus on discrete data, the entropy based discretization method is applied in order to deal with continuous attributes for the experiments. More concretely, a recursive entropy minimization heuristic is used for discretization and combined with the Minimum Description Length criterion to control the number of intervals produced over a continuous space [15].

4.1. Accuracy

Accuracy is one of the basic performance measures for classification algorithms. For a classifier,

Table 5
Sub-algorithm *addToCand*

```
addToCand(c, cand);
1.  begin
2.    find=0; count=1;
3.      for each candidate itemset c_i in cand do
4.      if c = c_i then
5.        c_i.wcount= c_i.wcount+1;
6.        c_i.lcount= c_i.lcount+1;
7.        find = 1;
8.      else
9.        if ancetedent(c) = ancetedent(c_i) then
10.       c_i.lcount = c_i.lcount+1;
11.       count = count + c_i.lcount;
12.     end if;
13.   end for;
14.   if find = 0 and (consequent of c is not equal to q) then
      //when c is not redundant//
15.     c is included in cand with c.lcount=count and
        c.wcount=1;
16.   end if;
17. end;
```

Table 6
Basic information of the 30 UCI datasets

	Dataset	Attributes	Number of attributes	Null value (Y/N)	Number of training data	Number of testing data
1	Anneal	Discrete, continuous	38	Y	598	300
2	Australian	Discrete, continuous	14	N	460	230
3	Auto	Discrete, continuous	26	Y	136	69
4	Breast	Continuous	10	Y	466	233
5	Cleve	Discrete, continuous	13	N	202	101
6	Crx	Discrete, continuous	15	N	490	200
7	Diabetes	Continuous	8	N	512	256
8	German	Discrete, continuous	20	N	666	33
9	Glass	Continuous	9	N	142	72
10	Heart	Continuous	13	N	180	90
11	Hepatitis	Discrete, continuous	19	Y	103	52
12	Horse	Discrete, continuous	22	Y	300	68
13	Hypothyroid	Discrete, continuous	29	Y	2514	1258
14	Ionosphere	Continuous	34	Y	234	117
15	Iris	Continuous	4	N	100	50
16	Labor	Discrete, continuous	16	Y	40	17
17	Led7	Discrete	7	N	200	3000
18	Lymph	Discrete	18	N	98	50
19	Pima	Continuous	8	N	512	256
20	Sick	Discrete, continuous	29	Y	2800	972
21	Sonar	Continuous	60	N	138	70
22	Tic-tac-toe	Discrete	9	N	638	320
23	Vehicle	Continuous	18	N	564	282
24	Waveform	Continuous	21	N	300	1000
25	Wine	Continuous	13	N	118	60
26	Zoo	Discrete	16	N	67	34
27	Balance	Continuous	4	N	416	209
28	Lenses	Discrete	4	N	16	8
29	Monk2	Discrete	6	N	169	432
30	Vote	Discrete	16	N	300	135

its classification accuracy is the ratio of the number of cases truly predicted by the classifier over the total number of cases in the test dataset, e.g.,

$$\text{Accuracy} = \frac{\text{num}(\text{test\_predicted} = \text{true})}{\text{num\_totaltest}} \times 100\%.$$

In this experiment, we compared GARC with C4.5 rule, CBA, NN and SVM classifiers based on the 30 UCI datasets. We obtained the accuracy results as shown in Tables 7, 8. It will be discussed in later subsections on how the settings of the thresholds are considered.

The results shown in Tables 7 and 8 indicate that the classification accuracy of GARC is satisfactory. On average, the accuracy of GARC seemed to be higher than that of the C4.5 rule and similar to that of CBA. Moreover the GARC classifier appeared to be more stable than CBA and C4.5 rule classifiers in terms of standard deviations of accuracy. These findings could be further justified by statistical significance tests. Moreover, Table 8 shows that the accuracy of GARC is lower than that of NN or SVM, and that the standard deviation of GARC is lower than that of NN and SVM. However, it is important to note that GARC, NN, SVM are not significantly different in accuracy.

Thus, we could test the significance of the accuracy mean difference for any two algorithms by approximately constructing a confidence interval at a given confidence level [6,18]. The testing results revealed that on average the accuracy of GARC was

Table 7
Algorithms' accuracy on C4.5, CBA and GARC

	Datasets	C4.5%	CBA%	GARC%
1	Anneal	88.70	98.00	89.30
2	Australian	87.00	86.96	87.39
3	Auto	62.70	72.46	71.32
4	Breast	95.70	96.57	94.85
5	Cleve	77.20	81.19	80.13
6	Crx	83.00	83.50	82.50
7	Diabetes	69.10	74.22	71.03
8	German	73.40	76.35	75.20
9	Glass	62.50	65.28	68.06
10	Heart	83.30	83.33	80.57
11	Hepatitis	80.80	76.92	86.69
12	Horse	85.30	80.88	75.00
13	Hypothyroid	99.20	98.20	94.79
14	Ionosphere	88.00	93.16	90.64
15	Iris	92.00	94.00	94.01
16	Labor	82.40	88.24	82.35
17	Led7	67.50	57.67	56.53
18	Lymph	70.00	84.00	77.56
19	Pima	76.60	76.17	73.83
20	Sick	99.00	96.50	93.83
21	Sonar	74.30	64.29	74.30
22	Tic-tac-toe	82.20	99.06	100.00
23	Vehicle	67.70	70.21	61.89
24	Waveform	70.40	75.66	71.15
25	Wine	85.00	86.67	83.46
26	Zoo	85.30	79.41	82.35
27	Balance	77.50	72.73	71.29
28	Lenses	62.50	62.50	75.32
29	Monk2	65.00	67.13	65.74
30	Vote	97.00	95.56	89.67
	Mean	79.68	81.23	80.03
	Derivation	1.23	1.40	1.15
	Standard deviation	11.09	11.84	10.72

GARC is running with default setting of $\alpha=0.01$, $\beta=0.7$.

not significantly different from that of CBA, C4.5, NN or SVM. These have been shown in Table 9.

In addition, two C4.5 extensions, namely C4.5 tree and C4.5 tree pruning [30], were tested on the same 30 datasets by means of confidence intervals, revealing that the accuracy of GARC was not significantly different from that of either C4.5 tree or C4.5 tree pruning at 95% confidence level. Furthermore, our test on the 30 datasets is, however, not supportive to the statement in Ref. [21] that the accuracy of the C4.5 rule is higher than that of either C4.5 tree or C4.5 tree pruning.

In summary, GARC is satisfactory in terms of accuracy, compared with CBA, C4.5-type, NN and SVM. Worthwhile to mention is that, compared with non-rule-based classifiers (e.g., NN and SVM), GARC produces a classifier in the form of explicit rules, which are often appealing for use and explanation to decision makers.

Table 8
Algorithms' accuracy on SVM, NN and GARC

	Datasets	SVM%	NN%	GARC%
1	Anneal	100.00	98.67	96.67
2	Australian	86.96	90.00	87.39
3	Auto	72.46	63.77	71.00
4	Breast	96.14	94.85	95.71
5	Cleve	82.18	81.19	81.19
6	Crx	82.50	85.00	85.50
7	Diabetes	73.83	78.13	74.61
8	German	71.86	75.15	76.35
9	Glass	79.17	73.61	68.06
10	Heart	88.88	87.78	88.00
11	Hepatitis	84.62	78.85	86.54
12	Horse	86.76	80.88	88.24
13	Hypothyroid	100.00	98.01	94.79
14	Ionosphere	96.58	94.87	94.85
15	Iris	94.00	96.00	96.00
16	Labor	100.00	94.12	82.35
17	Led7	68.97	67.73	67.47
18	Lymph	82.00	86.00	80.00
19	Pima	79.69	78.52	76.17
20	Sick	96.71	97.02	93.83
21	Sonar	88.57	78.57	74.30
22	Tic-tac-toe	99.38	97.50	100.00
23	Vehicle	79.08	78.37	67.36
24	Waveform	80.85	81.06	71.55
25	Wine	98.33	95.00	86.67
26	Zoo	88.24	88.24	85.29
27	Balance	99.04	92.34	73.21
28	Lenses	62.50	75.00	87.50
29	Monk2	84.72	100.00	74.54
30	Vote	97.78	99.26	96.30
	Mean	86.73	86.18	83.38
	Deviation	1.11	1.03	1.00
	Standard deviation	10.52	10.13	10.01

SVM classifier use LIBSVM software package available online at http://www.csie.ntu.edu.tw/~cjlin/libsvm Ref. [36]. For SVM, the parameters will largely affect the results. With default settings by LIBSVM, the average accuracy is 79.84. After parameters selection with 10-fold cross validation, SVM is running on the best situation.

NN classifier is using WEKA software package [35]. A 3-level Back-Propagation Model has been constructed, with the number of neurons set to 2, 4 and 8. The results are not sensitive to the number of neurons. NN is also running on the best situation. GARC is running on the best situation with corresponding α and β.

Table 9
Confidence intervals on the mean difference for accuracy of classifiers

	Confidence level %	Interval%	Significance
GARC–CBA	95	[−6.92,4.51]	No
	90	[−6.00,3.59]	No
GARC–C4.5 rule	95	[−5.17,5.87]	No
	90	[−4.28,4.98]	No
GARC–NN	95	[−8.71,1.74]	No
	90	[−7.87,0.90]	No
GARC–SVM	95	[−9.35,1.30]	No
	90	[−8.49,0.44]	No

4.2. GARC with pruning strategies

This subsection examines GARC's pruning strategy effectiveness in terms of accuracy, understandability (e.g., the number of rules generated) and computational efficiency (e.g., the number of candidate itemsets generated, execution time, etc.). By a pruning strategy we mean the strategy discussed in Section 2.4 and incorporated in the mining process. Since we are to study the GARC performance with and without the strategy incorporation, the dataset used needs to be expansible and adjustable in size and complexity. Apparently, the previously used 30 datasets can hardly serve this purpose. Hence, as proposed in Ref. [3] for similar experiments, a synthetic database is employed. Each transaction in the database has 9 attributes shown in Table 10. There are ten classification functions available to produce data distributions with varied complexities. IBM Research Center developed a series of classification functions of increasing complexity that used the above attributes to classify people into different groups [19]. Four of them are selected, which include the low-complexity (function 2), mid-complexity (function 5 and 8) and the most complex function 10. Specifically, function 10 is one of the hardest to characterize and could result in the highest classification errors (Table 11). The Data generator source is from Ref. [19].

Table 10
Attributes of the synthetic database

Attribute	Value
Salary	Uniformly distributed from 20000 to 150000
Commission	If salary ≥75000, commission=0 else uniformly distributed from 10000 to 75000
Age	Uniformly distributed from 20 to 80
Ed_level	Uniformly chosen from 0 to 4
Car	Make of the car, uniformly chosen from 1 to 20
Zipcode	Uniformly chosen from 9 available zipcodes
Housevalue	Uniformly distributed from $0.5*k*100000$ to $1.5*k*100000$, where $0 \le k \le 9$ and depends on the zipcodes
Years owned	Uniformly distributed from 1 to 30
Loan	Uniformly distributed from 0 to 500000

Since GARC works with categorical attributes, the non-categorical attributes were discretized first. We used a simple equal-width method for discretization. The interval width and the number of intervals are shown in Table 12.

The performances of GARC with and without those (pruning) strategies proposed in Section 2 are shown in Fig. 1. The findings indicated that GARC with the strategies was superior to that without the strategies in three respects, namely, computational efficiency (fewer candidate itemsets and shorter execution time as shown in Fig. 1a,b), understandability (fewer rules

Table 11
Functions' definitions

Function 2:
Class A: ((age<40) ^ (50k≤salary≤100k))∨
((40≤age<60) ^ (75k≤salary≤125k))∨
((age60) ^ (25k≤salary≤75k)).
Function 5:
Class A: ((age<40) ^
(((50k≤salary≤100k)) ? (100k≤loan≤300k) :
(200k≤loan≤400k))))∨
((40≤age<60)^
(((75k≤salary≤125k)) ? (200k≤loan≤400k) :∨
(300k≤loan≤500k))))∨
((age≥60)^
(((25k≤salary≤75k)) ? (300k≤loan≤500k) :
(100k≤loan≤300k))))

Function 8:
disposable = (0.67 × (salary + commission) – 5000 × elevel –20k)
Class A: disposable >0

Function 10
hyears < 20⇒ equity=0
hyears ≥ 20⇒ equity=0.1 hvalue (hyeares 20)
disposable = (0.67 (salary+commission) 5000 × elevel + 0.2 equity –10k)
Class A: disposable >0

* A ? B : C stands for a logic expression meaning that if A is TRUE then B, else C.

Table 12
Discretization of the attribute values

Attribute	Interval width	No. of intervals
Salary	25,000	6
Commission	10,000	7
Age	10	6
Ed_level	–	5
Car	–	20
Zipcode	–	9
Housevalue	100,000	14
Years owned	3	10
Loan	50,000	10

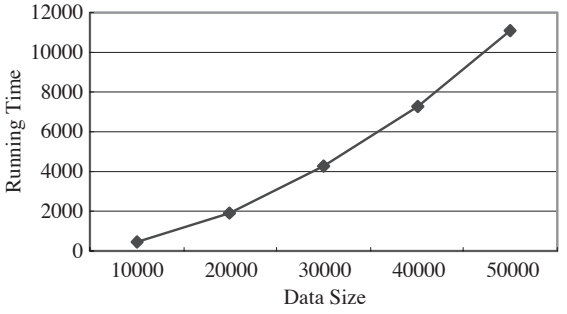


Fig. 2. Running time vs. data size.

as shown in Fig. 1c), and accuracy (similar rates as shown in Fig. 1d).

Further, Fig. 2 illustrates the execution time of GARC (e.g., for function 10) with the number of training samples increasing from 100,000 to 500,000, showing a near-linear computational complexity in time. This was also done for CBA and resulted in almost the same outcome.

4.3. Settings of minimal support and minimal confidence

As mentioned in previous subsections, the experiments were conducted with a setting of $\alpha=0.01$ and $\beta=0.7$ for min-support and min-confidence. In this section, we will discuss further the impact of such settings on the accuracy of GARC. With each of the same datasets, we could obtain the best setting of α and β in yielding the highest accuracy. Obviously, the best setting for one dataset is generally different from that for another dataset. To determine a single setting to be used in comparison for 30 datasets, we chose $\alpha=0.01$ and $\beta=0.7$, as they appeared quite often (e.g., about 19 times out of 30 for $\alpha=0.01$ and 14/30 for $\beta=0.7$). Table 13 shows the details.

Moreover, our experiments showed that as min-confidence increases, the accuracy would increase first then decrease. This may be because when min-confidence is too low, many useless rules will be generated, which will disturb the classifier. On the other hand, when min-confidence is too high, many actually meaningful rules will not be discovered, which will lead many transactions to being classified into the default class, resulting in lower accuracy. Generally, it could be found that the best performance of accuracy is around the situation where $\alpha=0.01$ and $\beta=0.7$.

4.4. Impact of the information gain measure in GARC

As discussed previously, GARC uses information gain to retrieve the best split attribute. In this section, some experimental results are given to show how this measure affects the accuracy and efficiency of GARC.

	With Strategies	Without Strategies
Function 2	21	228
Function 5	21	336
Function 8	87	144
Function 10	29	249

(a) Number of Candidate Itemsets

	With Strategies	Without Strategies
Function 2	13.24	179.10
Function 5	21	190
Function 8	14.74	16.25
Function 10	11.21	182.39

(b) Execution Time (sec.)

	With Strategies	Without Strategies
Function 2	5179	22231
Function 5	10302	26951
Function 8	11056	12000
Function 10	8435	28059

(c) Number of Rules

	With Strategies	Without Strategies
Function 2	0.84	0.83
Function 5	0.81	0.85
Function 8	0.90	0.90
Function 10	0.84	0.86

(d) Accuracy

Fig. 1. Performances of GARC with and without strategies.

Table 13
Settings of α and β vs. accuracy

	Highest accuracy			Highest accuracy at $\alpha=0.01$		Accuracy at $\alpha=0.01$, $\beta=0.7$ (%)
	%	α	β	%	β	
Anneal	96.67	0.01	0.95	96.67	0.95	89.33
Australian	87.39	0.01	0.7	87.39	0.7	87.39
Auto	71	0.01	0.7	71	0.7	71.07
Balance	73.21	0.01	0.85	73.21	0.85	71.29
Breast	95.71	0.01	0.95	95.71	0.95	94.85
Cleve	81.19	0.01	0.9	81.19	0.9	80.2
Crx	85.5	0.05	0.9	84.5	0.9	82.5
Diabetes	74.61	0.01	0.85	74.61	0.85	71.48
German	76.35	0.01	0.75	76.35	0.75	76.05
Glass	68.06	0.01	0.7	68.06	0.7	68.06
Heart	88	0.01	0.5	88	0.5	81.11
Hepatitis	86.54	0.01	0.7	86.54	0.7	86.54
Horse	88.24	0.01	0.85	88.24	0.85	75
Hypo	94.79	0.01	0.7	94.79	0.7	94.79
Ionosphere	94.87	0.01	0.95	94.87	0.95	91.45
Iris	96	0.01	1	96	1	94
Labor	82.35	0.01	0.7	82.35	0.7	82.35
Led7	67.47	0.02	0.5	66.33	0.5	57
Lenses	87.5	0.07	0.8	75	0.7	75
Lymph	80	0.02	0.8	78	0.8	78
Monk2	74.54	0.01	0.95	74.54	0.95	67.13
Pima	76.17	0.01	0.85	76.17	0.85	73.83
Sick	93.83	0.01	0.7	93.83	0.7	93.83
Sonar	74.3	0.01	0.7	74.3	0.7	74.3
Tic-tac-toe	100	0.01	0.7	100	0.7	100
Vehicle	67.36	0.01	0.7	67.36	0.7	67.36
Vote	96.3	0.01	0.95	96.3	0.95	89.67
Waveform	71.55	0.02	0.7	69.51	0.9	71
Wine	86.67	0.09	0.7	83.33	0.7	83.33
Zoo	85.29	0.05	0.95	82.35	0.7	82.35

Table 14
Accuracy by using information gain and not using information gain

Datasets	Information gain incorporated	Information gain not incorporated
Anneal	89.33	90
Australian	87.39	87.39
Auto	71.07	65.22
Balance	71.29	72.25
Breast	94.85	94.85
Cleve	80.2	68.32
Crx	82.5	81.5
Diabetes	71.48	67.97
German	76.05	72.55
Glass	68.06	66.67
Heart	81.11	81.11
Hepatitis	86.54	86.54
Horse	75	75
Hypo	94.79	94.79
Ionosphere	91.45	91.45
Iris	94	94
Labor	82.35	82.35
Led7	57	55.8
Lenses	75	75
Lymph	78	78
Monk2	67.13	67.13
Pima	73.83	73.83
Sick	93.83	93.83
Sonar	74.3	67.14
Tic-tac-toe	100	100
Vehicle	67.36	63.83
Vote	89.67	84.44
Waveform	71	71.98
Wine	83.33	83.33
Zoo	82.35	82.35
Mean	80.34	78.95
Standard deviation	10.35	11.29

The results revealed that the average accuracy with information gain was slightly higher than that without information gain. Statistically, the accuracy with information gain is significantly similar to that without information gain at both 95% and 90% confidence levels. These are shown in Tables 14 and 15.

In addition, considering the average number of rules and running times, the results revealed that information gain would lead to much fewer rules and less computational time remarkably. This is largely due to the fact that the number of candidate itemsets has been considerably reduced through introducing information gain in the mining process. On average, the number of rules with information gain was only around 39% of that without the gain, and the execution time with information gain was only around 3.2% of that without information gain, according to the experiment (shownin Table 16).

4.5. GARC and CBA

Though GARC and CBA are all based on association rule mining, they are different: CBA is basically

Table 15
Confidence intervals on the mean difference for accuracy

Information gain–No information gain	Confidence level%	Interval%	Significance
Accuracy	95	[−4.09, 6.87]	No
	90	[−3.21, 5.99]	No

of Apriori-type, whereas GARC is not. The main difference is that GARC combines rule generation with respective frequent itemset generation, making use of excluded itemsets in generating candidate itemsets. In addition, GARC uses information gain to reduce the number of candidate itemsets, which could then reduce the number of rules. Moreover, the number of rules could also be reduced using pruning/resolution strategies such that certain conflicting and redundant rules could be avoided, whereas the CBA algorithm itself will generate more rules. Table 17 tabulates the comparative results based on the 30 benchmarking datasets that were used previously. Clearly, GARC generated much fewer rules than CBA, providing better understandability (at similar levels of accuracy). This can easily be verified by statistical significance tests. On average, the number of rules generated by GARC was only around 4.3% of that by CBA, according to the experiment.

Table 16
Number of rules and running time by using information gain (IG) and not using information gain (NIG)

	Number of rules		Running time (s)	
	IG	NIG	IG	NIG
Anneal	72	85	21.422	7269.953
Australian	17	17	0.125	0.0160
Auto	650	2156	3785.860	116901.047
Balance	4	10	0.000000001	0.063
Breast	21	25	20.5	237.328
Cleve	23	37	253.564	1578.468
Crx	21	64	2.031	61.812
Diabetes	11	13	25.985	112.078
German	78	188	559.766	21953.438
Glass	17	21	2.062	12.687
Heart	12	12	0.000000001	0.00000001
Hepatitis	23	23	0.063	0.010
Horse	26	26	0.125	0.016
Hypo	48	48	0.265	0.094
Ionosphere	67	67	0.360	0.160
Iris	7	10	0.000000001	0.000000001
Labor	15	42	0.093	0.266
Led7	33	51	0.234	1.766
Lenses	12	13	0.000000001	0.000000001
Lymph	17	17	0.296	0.100
Monk2	2	2	0.000000001	0.000000001
Pima	6	6	6.016	33.594
Sick	56	56	0.281	0.172
Sonar	16	33	2.765	549.750
Tic-tac-toe	26	26	0.063	0.010
Vehicle	112	543	173.688	4852.828
Vote	32	96	0.937	49.969
Waveform	25	168	0.250	30.078
Wine	16	16	0.093	0.010
Zoo	90	151	2.390	60.313
Mean	51.83	134.07	161.97	5123.53
IG/NIG	39%		3.2%	

Table 17
Number of rules generated by GARC and CBA

	GARC	CBA
Anneal	72	533
Australian	17	1518
Auto	650	4505
Balance	4	147
Breast	21	21
Cleve	23	478
Crx	21	2686
Diabetes	11	40
German	78	1501
Glass	17	32
Heart	12	166
Hepatitis	23	700
Horse	26	988
Hypo	48	1557
Ionosphere	67	2891
Iris	7	14
Labor	15	52
Led7	33	533
Lenses	12	12
Lymph	17	2172
Monk2	2	397
Pima	6	21
Sick	56	1659
Sonar	16	883
Tic-tac-toe	26	200
Vehicle	112	3043
Vote	32	953
Waveform	25	3851
Wine	16	738
Zoo	90	2869
Mean	51.83	1205.33
GARC/CBA	4.3%	

5. Conclusions

Classification is one of the important issues in decision science and knowledge discovery. This paper has presented a new approach to discovering classification rules based on the concept of association rules. In doing so, the corresponding algorithm proposed, namely GARC, has integrated the generation

of itemsets and rules, and incorporated information gain and certain conflicts/redundancy resolution strategies into the mining process. Finally, a compact set could be derived. Compared with other classifiers (e.g., CBA, C4.5-type, NN and SVM classifiers), the new approach could achieve a similar level of accuracy. Moreover, the experimental results have shown the advantages of GARC over CBA in terms of number of rules, and over SVM/NN in terms of explicit rules for use and explanation by decision makers. Future studies are centering on explorations of other optimization strategies so as to further improve the mining efficiency.

Acknowledgements

This work was partly supported by the National Natural Science Foundation of China (79925001/70231010), and China's MOE Funds for Doctoral Programs (20020003095).

References

[1] R. Agrawal, R. Srikant, Fast algorithm for mining association rules, Proceeding of the 20th VLDB conference, Morgan Kaufmann, Santiago, Chile, 1994, pp. 487–499.
[2] R. Agrawal, T. Imielinski, A. Swami, Database mining: a performance perspective, IEEE Transaction on Knowledge and Data Engineering 5 (1993) 914–925.
[3] R. Agrawal, T. Imielinski, A. Swami, Mining association rules between sets of items in large databases, Proceeding of 1993 ACM-SIGMOD International Conference on Management of Data, ACM Press, Washington, D.C., 1993, pp. 207–216.
[4] K. Ali, K. Manganaris, R. Srikant, Partial classification using association rules, Proceeding of the Third International Conference on Knowledge Discovery and Data Mining, The AAAI Press, Newport Beach, California, 1997, pp. 115–118.
[5] K. Alsabti, S. Ranka, V. Singh, CLOUDS: a decision tree classifier for large datasets, in: R. Agrawal, P. Stolorz, G. Piatetsky-Shapiro (Eds.), Proceeding of the Fourth Int. Conference on Knowledge Discovery and Data Mining, AAAI Press, Newport Beach, California, 1998, pp. 2–8 (New York, New York).
[6] D. Bertsimas, R.M. Freund, Data, Model, and Decisions: the Fundamentals of Management Science, South-Western College Publishing, 2000.
[7] L. Breiman, Classification and Regression trees, Wadsworth, Belmont, 1984.
[8] S. Brin, R. Motwani, J. Ullman, Dynamic itemset counting and implication rules for market basket data, Proceeding of 1997 ACM-SIGMOD International Conference on Management of Data, ACM Press, Tucson, Arizona, 1997, pp. 255–264.
[9] J. Catlett, Megainduction: Machine Learning on Very Large Databases. PhD thesis, University of Sydney, 1991.
[10] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines, 2001.
[11] G.Q. Chen, Q. Wei, Fuzzy association rules and the extended mining algorithms, Information Sciences 147 (2002) 201–228.
[12] G.Q. Chen, Q. Wei, E. Kerre, Fuzzy data mining: discovery of fuzzy generalized association rules, in: G. Bordogna, G. Pasi (Eds.), Recent Research Issues on the Management of Fuzziness in Databases, Physica-Verlag (Springer), 1999.
[13] G.Q. Chen, Q. Wei, D. Liu, G. Wets, Simple association rules (SAR) and the SAR-based rule discovery, Computers and Industrial Engineering 43 (2002) 721–733.
[14] R. Duda, P. Hart, Pattern Classification and Scene Analysis, John Wiley and Sons, 1973.
[15] U.M. Fayyad, K.B. Irani, Multi-interval discretization of continuous-valued attributes for classification learning, Proceedings of the 13th International Joint Conference on Artificial Intelligence, 1993, pp. 1022–1027.
[16] N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifier, Machine Learning 29 (1997) 131–163.
[17] T. Fukuda, Y. Morimoto, S. Morishita, Data mining using two-dimensional optimized association rules: scheme, algorithms, and visualization, Proc. of the 1996 ACM-SIGMOD Int'l Conf. on the Management of Data, 1996, pp. 12–13.
[18] Harshbarger,Thad R., Introductory Statistics: A Decision Map, Second edition. The City College, City University of New York, P 376, Macmillan Publishing Co., Inc. New York, Collier Macmillan Publishers, London, 1977.
[19] http://www.almaden.ibm.com/software/quest/Resources/datasets/syndata.html#classSynData.
[20] W. Li, J. Han, J. Pei, CMAR: accurate and efficient classification based on multiple class-association rules, ICDM 2001, IEEE Computer Society, San Jose, California, 2001, pp. 369–376.
[21] B. Liu, W. Hsu, Y. Ma, Integrity classification and association rule mining, in: R. Agrawal, P. Stolorz, G. Piatetsky-Shapiro (Eds.), Proceeding of the Fourth Int. Conference on Knowledge Discovery and Data Mining, AAAI Press, New York, New York, 1998, pp. 80–86.
[22] H.Y. Liu, J. Chen, G. Chen, Mining insightful classification rules directly and efficiently, Proceeding of the 1999 IEEE International Conference on Systems Man and Cybernetics, IEEE Computer Society, Tokyo, 1999, pp. 911–916.
[23] B. Liu, Y. Ma, C. Wong, Classification using association rules: weaknesses and enhancements, in: Vipin Kumar, et al., (Eds.), Data Mining for Scientific and Engineering Applications, 2001, p. 591.
[24] H. Lu, H. Liu, Decision tables: Scalable classification exploring RDBMS capabilities, Proceeding of the 26th International Conference on Very Large Databases, Morgan Kaufmann, Cairo, Egypt, 2000, pp. 373–384.
[25] M. Mehta, R. Agrawal, J. Rissanen, SLIQ: A fast scalable classifier for data mining, Proceeding of the Fifth Interna-

tional Conference on Extending Database Technology, Springer, Avignon, France, 1996, pp. 18–32.
[26] D. Meretakis, B. Wüthrich, Extending naïve Bayes classifiers using long itemsets, Proceedings of 5th International Conference on Knowledge Discovery and Data Mining, San Diego, California, August 1999, 1999.
[27] C.J. Merz, P. Murphy, UCI Repository of Machine Learning Databases, 1996 (http://www.cs.uci.edu/~mlearn/MLRepository.html).
[28] A. Mueller, Fast Sequential and Parallel Algorithms for Association Rule Mining: A Comparison, 1995 (CS-TR-3515).
[29] G. Piatetsky-Shapiro, U. Fayyad, P. Smyth, From data mining to knowledge discovery, in: G. Piatetsky-Shapiro, U. Fayyad, P. Smyth (Eds.), An Overview. Advances in Knowledge Discovery and Data Mining, AAAI/MIT press, 1996, pp. 1–35.
[30] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, 1993.
[31] J. Roberto, J. Bayardo, Brute-force mining of high-confidence classification rules, Proceeding of the Third International Conference on Knowledge Discovery and Data Mining, AAAI Press, Newport Beach, California, 1997.
[32] J. Roberto, Bayardo Jr., R. Agrawal, D. Gunopulos, Constraint-based rule mining in large dense databases, Proc. of the 15th International Conference on Data Engineering, 1999, pp. 188–197.
[33] J. Shafer, R. Agrawal, M. Mehta, SPRINT: A scalable parallel classifier for data mining, Proceeding of the 22nd VLDB Conference, Morgan Kaufmann, India, 1996, pp. 544–555.
[34] R. Srikant, Q. Vu, R. Agrawal, Mining association rules with item constraints, Proc. of the 3rd Int'l Conference on Knowledge Discovery in Databases and Data Mining, AAAI Press, Newport Beach, California, USA, 1997.
[35] S. Weiss, C. Kulikowski, Computer Systems that Learn: Classification and Prediction Methods from Statistics, Neural Nets, Machine Learning, and Expert Systems, Morgan Kaufman, 1991.
[36] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, San Francisco, USA, 2000.
[37] M.J. Zaki, S. Parthasarathy, M. Ogihara, W. Li, New algorithms for fast discovery of association rules, American Association for Artificial Intelligence (1999).

Guoqing Chen received his PhD from the Catholic University of Leuven (K.U. Leuven, Belgium) and now is Professor of Information Systems at School of Economics and Management, Tsinghua University (Beijing, China). His research interests include Information Systems Management, Business Intelligence and Decision Support, and Soft Computing. Dr. Chen is member of ACM (SIGMOD and SIGKDD) and AIS and has wide publications internationally including a monographic book on data modeling published by Kluwer Academic Publishers (Boston, 1998).

Hongyan Liu has received her PhD in Management Science from Tsinghua University, China. She is an associate professor in the Management Science and Engineering Department at Tsinghua University. Her current research interests include database and information system, data warehouse, knowledge discovery in database and bioinformatics.

Lan Yu has received his bachelor's degree in Management Information Systems from the School of Economics and Management, Tsinghua University. He is a PhD candidate student in Management Science and Engineering Department at SEM. His current research interests focus on supervised learning (e.g., classification, reinforcement learning) and mainly apply them in finance and traffic domain.

Qiang Wei has received his PhD in Management Science from School of Economics and Management in 2003. He is an assistant professor in the Department of Management Science and Engineering, School of Economics and Management, Tsinghua University, China. His current research interests include knowledge discovery, data mining techniques, management information systems, system simulations. He has been a lead author for papers that have appeared in *Journal of Information Sciences*, *International Journal of Intelligent Systems*, and *Journal of Computer and Industrial Engineering*.

Xing Zhang has received his bachelor's degree in Management Information Systems from the School of Economics and Management, Tsinghua University. He is a PhD candidate student in Management Science and Engineering Department at SEM. His research interests focus on classification, reinforcement learning, and data mining techniques.

XGBoost: A Scalable Tree Boosting System

Tianqi Chen
University of Washington
tqchen@cs.washington.edu

Carlos Guestrin
University of Washington
guestrin@cs.washington.edu

ABSTRACT

Tree boosting is a highly effective and widely used machine learning method. In this paper, we describe a scalable end-to-end tree boosting system called XGBoost, which is used widely by data scientists to achieve state-of-the-art results on many machine learning challenges. We propose a novel sparsity-aware algorithm for sparse data and weighted quantile sketch for approximate tree learning. More importantly, we provide insights on cache access patterns, data compression and sharding to build a scalable tree boosting system. By combining these insights, XGBoost scales beyond billions of examples using far fewer resources than existing systems.

Keywords

Large-scale Machine Learning

1. INTRODUCTION

Machine learning and data-driven approaches are becoming very important in many areas. Smart spam classifiers protect our email by learning from massive amounts of spam data and user feedback; advertising systems learn to match the right ads with the right context; fraud detection systems protect banks from malicious attackers; anomaly event detection systems help experimental physicists to find events that lead to new physics. There are two important factors that drive these successful applications: usage of effective (statistical) models that capture the complex data dependencies and scalable learning systems that learn the model of interest from large datasets.

Among the machine learning methods used in practice, gradient tree boosting [10][1] is one technique that shines in many applications. Tree boosting has been shown to give state-of-the-art results on many standard classification benchmarks [16]. LambdaMART [5], a variant of tree boosting for ranking, achieves state-of-the-art result for ranking problems. Besides being used as a stand-alone predictor, it is also incorporated into real-world production pipelines for ad click through rate prediction [15]. Finally, it is the de-facto choice of ensemble method and is used in challenges such as the Netflix prize [3].

In this paper, we describe XGBoost, a scalable machine learning system for tree boosting. The system is available as an open source package[2]. The impact of the system has been widely recognized in a number of machine learning and data mining challenges. Take the challenges hosted by the machine learning competition site Kaggle for example. Among the 29 challenge winning solutions [3] published at Kaggle's blog during 2015, 17 solutions used XGBoost. Among these solutions, eight solely used XGBoost to train the model, while most others combined XGBoost with neural nets in ensembles. For comparison, the second most popular method, deep neural nets, was used in 11 solutions. The success of the system was also witnessed in KDDCup 2015, where XGBoost was used by every winning team in the top-10. Moreover, the winning teams reported that ensemble methods outperform a well-configured XGBoost by only a small amount [1].

These results demonstrate that our system gives state-of-the-art results on a wide range of problems. Examples of the problems in these winning solutions include: store sales prediction; high energy physics event classification; web text classification; customer behavior prediction; motion detection; ad click through rate prediction; malware classification; product categorization; hazard risk prediction; massive online course dropout rate prediction. While domain dependent data analysis and feature engineering play an important role in these solutions, the fact that XGBoost is the consensus choice of learner shows the impact and importance of our system and tree boosting.

The most important factor behind the success of XGBoost is its scalability in all scenarios. The system runs more than ten times faster than existing popular solutions on a single machine and scales to billions of examples in distributed or memory-limited settings. The scalability of XGBoost is due to several important systems and algorithmic optimizations. These innovations include: a novel tree learning algorithm is for handling *sparse data*; a theoretically justified weighted quantile sketch procedure enables handling instance weights in approximate tree learning. Parallel and distributed computing makes learning faster which enables quicker model exploration. More importantly, XGBoost exploits out-of-core

[1]Gradient tree boosting is also known as gradient boosting machine (GBM) or gradient boosted regression tree (GBRT)

[2]https://github.com/dmlc/xgboost

[3]Solutions come from of top-3 teams of each competitions.

KDD '16, August 13-17, 2016, San Francisco, CA, USA

ACM ISBN .

DOI:

computation and enables data scientists to process hundred millions of examples on a desktop. Finally, it is even more exciting to combine these techniques to make an end-to-end system that scales to even larger data with the least amount of cluster resources. The major contributions of this paper is listed as follows:

- We design and build a highly scalable end-to-end tree boosting system.
- We propose a theoretically justified weighted quantile sketch for efficient proposal calculation.
- We introduce a novel sparsity-aware algorithm for parallel tree learning.
- We propose an effective cache-aware block structure for out-of-core tree learning.

While there are some existing works on parallel tree boosting [22, 23, 19], the directions such as out-of-core computation, cache-aware and sparsity-aware learning have not been explored. More importantly, an end-to-end system that combines all of these aspects gives a novel solution for real-world use-cases. This enables data scientists as well as researchers to build powerful variants of tree boosting algorithms [7, 8]. Besides these major contributions, we also make additional improvements in proposing a regularized learning objective, which we will include for completeness.

The remainder of the paper is organized as follows. We will first review tree boosting and introduce a regularized objective in Sec. 2. We then describe the split finding methods in Sec. 3 as well as the system design in Sec. 4, including experimental results when relevant to provide quantitative support for each optimization we describe. Related work is discussed in Sec. 5. Detailed end-to-end evaluations are included in Sec. 6. Finally we conclude the paper in Sec. 7.

2. TREE BOOSTING IN A NUTSHELL

We review gradient tree boosting algorithms in this section. The derivation follows from the same idea in existing literatures in gradient boosting. Specicially the second order method is originated from Friedman et al. [12]. We make minor improvements in the reguralized objective, which were found helpful in practice.

2.1 Regularized Learning Objective

For a given data set with n examples and m features $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}$ ($|\mathcal{D}| = n, \mathbf{x}_i \in \mathbb{R}^m, y_i \in \mathbb{R}$), a tree ensemble model (shown in Fig. 1) uses K additive functions to predict the output.

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \quad f_k \in \mathcal{F}, \tag{1}$$

where $\mathcal{F} = \{f(\mathbf{x}) = w_{q(\mathbf{x})}\}(q : \mathbb{R}^m \to T, w \in \mathbb{R}^T)$ is the space of regression trees (also known as CART). Here q represents the structure of each tree that maps an example to the corresponding leaf index. T is the number of leaves in the tree. Each f_k corresponds to an independent tree structure q and leaf weights w. Unlike decision trees, each regression tree contains a continuous score on each of the leaf, we use w_i to represent score on i-th leaf. For a given example, we will use the decision rules in the trees (given by q) to classify

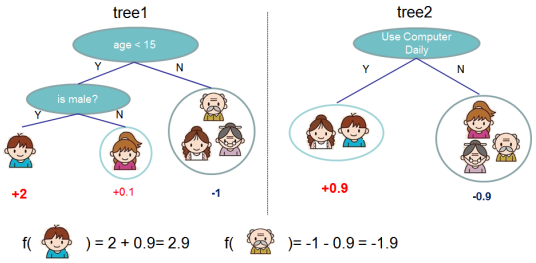


Figure 1: Tree Ensemble Model. The final prediction for a given example is the sum of predictions from each tree.

it into the leaves and calculate the final prediction by summing up the score in the corresponding leaves (given by w). To learn the set of functions used in the model, we minimize the following *regularized* objective.

$$\begin{aligned} \mathcal{L}(\phi) &= \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \\ \text{where } \Omega(f) &= \gamma T + \frac{1}{2}\lambda \|w\|^2 \end{aligned} \tag{2}$$

Here l is a differentiable convex loss function that measures the difference between the prediction $\hat{y}_i$ and the target y_i. The second term Ω penalizes the complexity of the model (i.e., the regression tree functions). The additional regularization term helps to smooth the final learnt weights to avoid over-fitting. Intuitively, the regularized objective will tend to select a model employing simple and predictive functions. A similar regularization technique has been used in Regularized greedy forest (RGF) [25] model. Our objective and the corresponding learning algorithm is simpler than RGF and easier to parallelize. When the regularization parameter is set to zero, the objective falls back to the traditional gradient tree boosting.

2.2 Gradient Tree Boosting

The tree ensemble model in Eq. (2) includes functions as parameters and cannot be optimized using traditional optimization methods in Euclidean space. Instead, the model is trained in an additive manner. Formally, let $\hat{y}_i^{(t)}$ be the prediction of the i-th instance at the t-th iteration, we will need to add f_t to minimize the following objective.

$$\mathcal{L}^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y_i}^{(t-1)} + f_t(\mathbf{x}_i)) + \Omega(f_t)$$

This means we greedily add the f_t that most improves our model according to Eq. (2). Second-order approximation can be used to quickly optimize the objective in the general setting [12].

$$\mathcal{L}^{(t)} \simeq \sum_{i=1}^{n} [l(y_i, \hat{y}^{(t-1)}) + g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t)$$

where $g_i = \partial_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ and $h_i = \partial^2_{\hat{y}^{(t-1)}} l(y_i, \hat{y}^{(t-1)})$ are first and second order gradient statistics on the loss function. We can remove the constant terms to obtain the following simplified objective at step t.

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^{n} [g_i f_t(\mathbf{x}_i) + \frac{1}{2} h_i f_t^2(\mathbf{x}_i)] + \Omega(f_t) \tag{3}$$

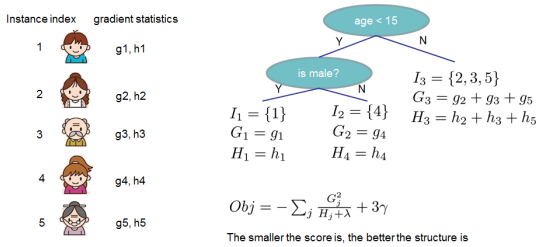


Figure 2: Structure Score Calculation. We only need to sum up the gradient and second order gradient statistics on each leaf, then apply the scoring formula to get the quality score.

Define $I_j = \{i|q(\mathbf{x}_i) = j\}$ as the instance set of leaf j. We can rewrite Eq (3) by expanding Ω as follows

$$\begin{aligned}\tilde{\mathcal{L}}^{(t)} &= \sum_{i=1}^{n}[g_i f_t(\mathbf{x}_i) + \frac{1}{2}h_i f_t^2(\mathbf{x}_i)] + \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2 \\ &= \sum_{j=1}^{T}[(\sum_{i\in I_j} g_i)w_j + \frac{1}{2}(\sum_{i\in I_j} h_i + \lambda)w_j^2] + \gamma T\end{aligned} \tag{4}$$

For a fixed structure $q(\mathbf{x})$, we can compute the optimal weight w_j^* of leaf j by

$$w_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda}, \tag{5}$$

and calculate the corresponding optimal value by

$$\tilde{\mathcal{L}}^{(t)}(q) = -\frac{1}{2}\sum_{j=1}^{T}\frac{(\sum_{i\in I_j} g_i)^2}{\sum_{i\in I_j} h_i + \lambda} + \gamma T. \tag{6}$$

Eq (6) can be used as a scoring function to measure the quality of a tree structure q. This score is like the impurity score for evaluating decision trees, except that it is derived for a wider range of objective functions. Fig. 2 illustrates how this score can be calculated.

Normally it is impossible to enumerate all the possible tree structures q. A greedy algorithm that starts from a single leaf and iteratively adds branches to the tree is used instead. Assume that I_L and I_R are the instance sets of left and right nodes after the split. Lettting $I = I_L \cup I_R$, then the loss reduction after the split is given by

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{(\sum_{i\in I_L} g_i)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{(\sum_{i\in I_R} g_i)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{(\sum_{i\in I} g_i)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma \tag{7}$$

This formula is usually used in practice for evaluating the split candidates.

2.3 Shrinkage and Column Subsampling

Besides the regularized objective mentioned in Sec. 2.1, two additional techniques are used to further prevent overfitting. The first technique is shrinkage introduced by Friedman [11]. Shrinkage scales newly added weights by a factor η after each step of tree boosting. Similar to a learning rate in tochastic optimization, shrinkage reduces the influence of each individual tree and leaves space for future trees to improve the model. The second technique is column (feature) subsampling. This technique is used in RandomForest [4, 13], It is implemented in a commercial software TreeNet [4] for gradient boosting, but is not implemented in existing opensource packages. According to user feedback, using column sub-sampling prevents over-fitting even more so than the traditional row sub-sampling (which is also supported). The usage of column sub-samples also speeds up computations of the parallel algorithm described later.

Algorithm 1: Exact Greedy Algorithm for Split Finding

Input: I, instance set of current node
Input: d, feature dimension
$gain \leftarrow 0$
$G \leftarrow \sum_{i\in I} g_i$, $H \leftarrow \sum_{i\in I} h_i$
for $k = 1$ ***to*** m **do**
  $G_L \leftarrow 0$, $H_L \leftarrow 0$
  for j *in sorted(I, by* $\mathbf{x}_{jk}$*)* **do**
    $G_L \leftarrow G_L + g_j$, $H_L \leftarrow H_L + h_j$
    $G_R \leftarrow G - G_L$, $H_R \leftarrow H - H_L$
    $score \leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  end
end
Output: Split with max score

Algorithm 2: Approximate Algorithm for Split Finding

for $k = 1$ ***to*** m **do**
  Propose $S_k = \{s_{k1}, s_{k2}, \cdots s_{kl}\}$ by percentiles on feature k.
  Proposal can be done per tree (global), or per split(local).
end
for $k = 1$ ***to*** m **do**
  $G_{kv} \leftarrow= \sum_{j\in\{j|s_{k,v}\geq \mathbf{x}_{jk} > s_{k,v-1}\}} g_j$
  $H_{kv} \leftarrow= \sum_{j\in\{j|s_{k,v}\geq \mathbf{x}_{jk} > s_{k,v-1}\}} h_j$
end
Follow same step as in previous section to find max score only among proposed splits.

3. SPLIT FINDING ALGORITHMS

3.1 Basic Exact Greedy Algorithm

One of the key problems in tree learning is to find the best split as indicated by Eq (7). In order to do so, a split finding algorithm enumerates over all the possible splits on all the features. We call this the *exact greedy algorithm*. Most existing single machine tree boosting implementations, such as scikit-learn [20], R's gbm [21] as well as the single machine version of XGBoost support the exact greedy algorithm. The exact greedy algorithm is shown in Alg. 1. It is computationally demanding to enumerate all the possible splits for continuous features. In order to do so efficiently, the algorithm must first sort the data according to feature values and visit the data in sorted order to accumulate the gradient statistics for the structure score in Eq (7).

3.2 Approximate Algorithm

The exact greedy algorithm is very powerful since it enumerates over all possible splitting points greedily. However, it is impossible to efficiently do so when the data does not fit entirely into memory. Same problem also arises in the dis-

[4]https://www.salford-systems.com/products/treenet

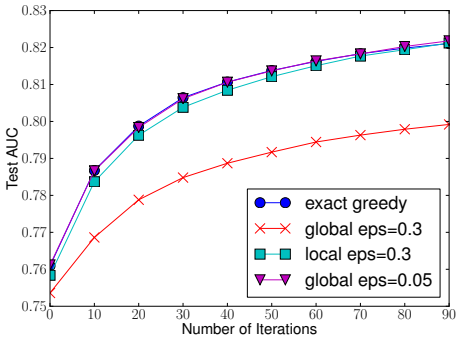


Figure 3: Comparison of test AUC convergence on Higgs 10M dataset. The eps parameter corresponds to the accuracy of the approximate sketch. This roughly translates to 1 / eps buckets in the proposal. We find that local proposals require fewer buckets, because it refine split candidates.

tributed setting. To support effective gradient tree boosting in these two settings, an approximate algorithm is needed.

We summarize an approximate framework, which resembles the ideas proposed in past literatures [17, 2, 22], in Alg. 2. To summarize, the algorithm first proposes candidate splitting points according to percentiles of feature distribution (a specific criteria will be given in Sec. 3.3). The algorithm then maps the continuous features into buckets split by these candidate points, aggregates the statistics and finds the best solution among proposals based on the aggregated statistics.

There are two variants of the algorithm, depending on when the proposal is given. The global variant proposes all the candidate splits during the initial phase of tree construction, and uses the same proposals for split finding at all levels. The local variant re-proposes after each split. The global method requires less proposal steps than the local method. However, usually more candidate points are needed for the global proposal because candidates are not refined after each split. The local proposal refines the candidates after splits, and can potentially be more appropriate for deeper trees. A comparison of different algorithms on a Higgs boson dataset is given by Fig. 3. We find that the local proposal indeed requires fewer candidates. The global proposal can be as accurate as the local one given enough candidates.

Most existing approximate algorithms for distributed tree learning also follow this framework. Notably, it is also possible to directly construct approximate histograms of gradient statistics [22]. It is also possible to use other variants of binning strategies instead of quantile [17]. Quantile strategy benefit from being distributable and recomputable, which we will detail in next subsection. From Fig. 3, we also find that the quantile strategy can get the same accuracy as exact greedy given reasonable approximation level.

Our system efficiently supports exact greedy for the single machine setting, as well as approximate algorithm with both local and global proposal methods for all settings. Users can freely choose between the methods according to their needs.

3.3 Weighted Quantile Sketch

One important step in the approximate algorithm is to propose candidate split points. Usually percentiles of a feature are used to make candidates distribute evenly on the data. Formally, let multi-set $\mathcal{D}_k = \{(x_{1k}, h_1), (x_{2k}, h_2) \cdots (x_{nk}, h_n)\}$ represent the k-th feature values and second order gradient statistics of each training instances. We can define a rank functions $r_k : \mathbb{R} \to [0, +\infty)$ as

$$r_k(z) = \frac{1}{\sum_{(x,h)\in\mathcal{D}_k} h} \sum_{(x,h)\in\mathcal{D}_k, x<z} h, \tag{8}$$

which represents the proportion of instances whose feature value k is smaller than z. The goal is to find candidate split points $\{s_{k1}, s_{k2}, \cdots s_{kl}\}$, such that

$$|r_k(s_{k,j}) - r_k(s_{k,j+1})| < \epsilon, \quad s_{k1} = \min_i \mathbf{x}_{ik}, s_{kl} = \max_i \mathbf{x}_{ik}. \tag{9}$$

Here ϵ is an approximation factor. Intuitively, this means that there is roughly $1/\epsilon$ candidate points. Here each data point is weighted by h_i. To see why h_i represents the weight, we can rewrite Eq (3) as

$$\sum_{i=1}^{n} \frac{1}{2} h_i (f_t(\mathbf{x}_i) - g_i/h_i)^2 + \Omega(f_t) + constant,$$

which is exactly weighted squared loss with labels g_i/h_i and weights h_i. For large datasets, it is non-trivial to find candidate splits that satisfy the criteria. When every instance has equal weights, an existing algorithm called quantile sketch [14, 24] solves the problem. However, there is no existing quantile sketch for the weighted datasets. Therefore, most existing approximate algorithms either resorted to sorting on a random subset of data which have a chance of failure or heuristics that do not have theoretical guarantee.

To solve this problem, we introduced a novel distributed weighted quantile sketch algorithm that can handle weighted data with a *provable theoretical guarantee*. The general idea is to propose a data structure that supports *merge* and *prune* operations, with each operation proven to maintain a certain accuracy level. A detailed description of the algorithm as well as proofs are given in the appendix.

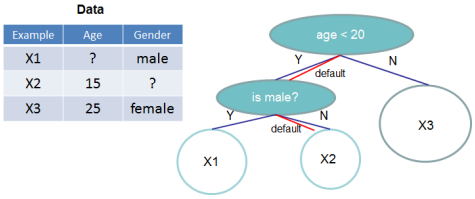


Figure 4: Tree structure with default directions. An example will be classified into the default direction when the feature needed for the split is missing.

3.4 Sparsity-aware Split Finding

In many real-world problems, it is quite common for the input $\mathbf{x}$ to be sparse. There are multiple possible causes for sparsity: 1) presence of missing values in the data; 2) frequent zero entries in the statistics; and, 3) artifacts of feature engineering such as one-hot encoding. It is important to make the algorithm aware of the sparsity pattern in the data. In order to do so, we propose to add a default direction in each tree node, which is shown in Fig. 4. When a value is missing in the sparse matrix $\mathbf{x}$, the instance is classified into the default direction. There are two choices

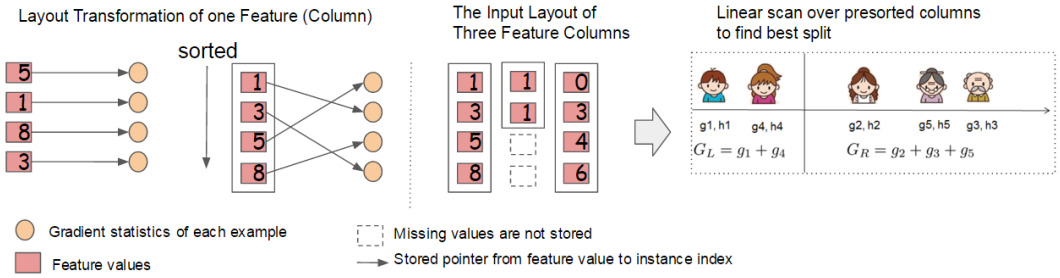


Figure 6: Block structure for parallel learning. Each column in a block is sorted by the corresponding feature value. A linear scan over one column in the block is sufficient to enumerate all the split points.

Algorithm 3: Sparsity-aware Split Finding

Input: I, instance set of current node
Input: $I_k = \{i \in I | x_{ik} \neq \text{missing}\}$
Input: d, feature dimension
Also applies to the approximate setting, only collect statistics of non-missing entries into buckets
$gain \leftarrow 0$
$G \leftarrow \sum_{i \in I} g_i, H \leftarrow \sum_{i \in I} h_i$
for $k = 1$ ***to*** m **do**
  // enumerate missing value goto right
  $G_L \leftarrow 0,\ H_L \leftarrow 0$
  for j *in sorted*$(I_k,$ *ascent order by* $\mathbf{x}_{jk})$ **do**
    $G_L \leftarrow G_L + g_j,\ H_L \leftarrow H_L + h_j$
    $G_R \leftarrow G - G_L,\ H_R \leftarrow H - H_L$
    $score \leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  end
  // enumerate missing value goto left
  $G_R \leftarrow 0,\ H_R \leftarrow 0$
  for j *in sorted*$(I_k,$ *descent order by* $\mathbf{x}_{jk})$ **do**
    $G_R \leftarrow G_R + g_j,\ H_R \leftarrow H_R + h_j$
    $G_L \leftarrow G - G_R,\ H_L \leftarrow H - H_R$
    $score \leftarrow \max(score, \frac{G_L^2}{H_L+\lambda} + \frac{G_R^2}{H_R+\lambda} - \frac{G^2}{H+\lambda})$
  end
end
Output: Split and default directions with max gain

of default direction in each branch. The optimal default directions are learnt from the data. The algorithm is shown in Alg. 3. The key improvement is to only visit the non-missing entries I_k. The presented algorithm treats the non-presence as a missing value and learns the best direction to handle missing values. The same algorithm can also be applied when the non-presence corresponds to a user specified value by limiting the enumeration only to consistent solutions.

To the best of our knowledge, most existing tree learning algorithms are either only optimized for dense data, or need specific procedures to handle limited cases such as categorical encoding. XGBoost handles all sparsity patterns in a unified way. More importantly, our method exploits the sparsity to make computation complexity linear to number of non-missing entries in the input. Fig. 5 shows the comparison of sparsity aware and a naive implementation on an Allstate-10K dataset (description of dataset given in Sec. 6). We find that the sparsity aware algorithm runs 50 times faster than the naive version. This confirms the importance of the sparsity aware algorithm.

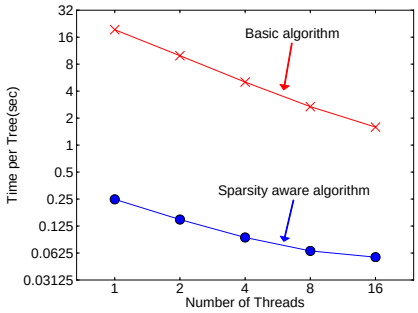


Figure 5: Impact of the sparsity aware algorithm on Allstate-10K. The dataset is sparse mainly due to one-hot encoding. The sparsity aware algorithm is more than 50 times faster than the naive version that does not take sparsity into consideration.

4. SYSTEM DESIGN

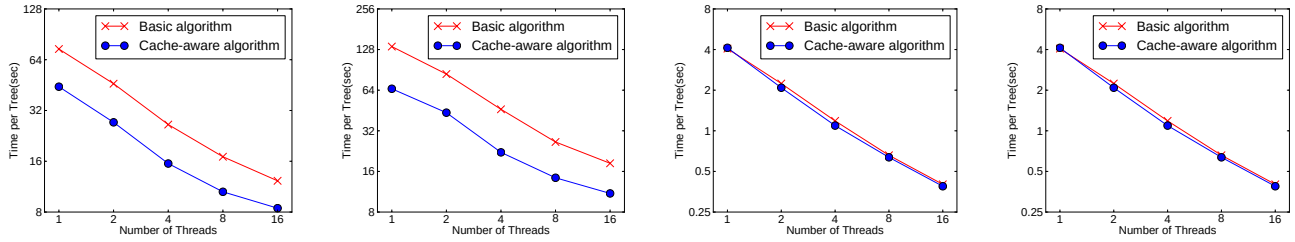
4.1 Column Block for Parallel Learning

The most time consuming part of tree learning is to get the data into sorted order. In order to reduce the cost of sorting, we propose to store the data in in-memory units, which we called *block*. Data in each block is stored in the compressed column (CSC) format, with each column sorted by the corresponding feature value. This input data layout only needs to be computed once before training, and can be reused in later iterations.

In the exact greedy algorithm, we store the entire dataset in a single block and run the split search algorithm by linearly scanning over the pre-sorted entries. We do the split finding of all leaves collectively, so one scan over the block will collect the statistics of the split candidates in all leaf branches. Fig. 6 shows how we transform a dataset into the format and find the optimal split using the block structure.

The block structure also helps when using the approximate algorithms. Multiple blocks can be used in this case, with each block corresponding to subset of rows in the dataset. Different blocks can be distributed across machines, or stored on disk in the out-of-core setting. Using the sorted structure, the quantile finding step becomes a *linear scan* over the sorted columns. This is especially valuable for local proposal algorithms, where candidates are generated frequently at each branch. The binary search in histogram aggregation also becomes a linear time merge style algorithm.

Collecting statistics for each column can be *parallelized*, giving us a parallel algorithm for split finding. Importantly, the column block structure also supports column subsampling, as it is easy to select a subset of columns in a block.



(a) Allstate 10M (b) Higgs 10M (c) Allstate 1M (d) Higgs 1M

Figure 7: Impact of cache-aware prefetching in exact greedy algorithm. We find that the cache-miss effect impacts the performance on the large datasets (10 million instances). Using cache aware prefetching improves the performance by factor of two when the dataset is large.

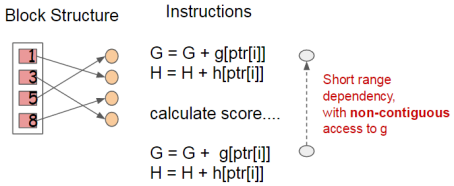


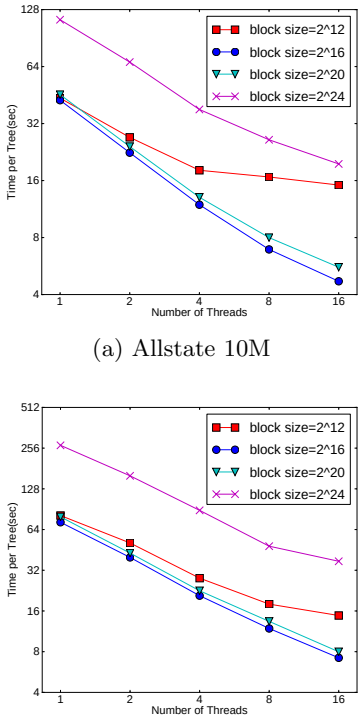
Figure 8: Short range data dependency pattern that can cause stall due to cache miss.

Time Complexity Analysis Let d be the maximum depth of the tree and K be total number of trees. For the exact greedy algorithm, the time complexity of original spase aware algorithm is $O(Kd\|\mathbf{x}\|_0 \log n)$. Here we use $\|\mathbf{x}\|_0$ to denote number of non-missing entries in the training data. On the other hand, tree boosting on the block structure only cost $O(Kd\|\mathbf{x}\|_0 + \|\mathbf{x}\|_0 \log n)$. Here $O(\|\mathbf{x}\|_0 \log n)$ is the one time preprocessing cost that can be amortized. This analysis shows that the block structure helps to save an additional $\log n$ factor, which is significant when n is large. For the approximate algorithm, the time complexity of original algorithm with binary search is $O(Kd\|\mathbf{x}\|_0 \log q)$. Here q is the number of proposal candidates in the dataset. While q is usually between 32 and 100, the log factor still introduces overhead. Using the block structure, we can reduce the time to $O(Kd\|\mathbf{x}\|_0 + \|\mathbf{x}\|_0 \log B)$, where B is the maximum number of rows in each block. Again we can save the additional $\log q$ factor in computation.

4.2 Cache-aware Access

While the proposed block structure helps optimize the computation complexity of split finding, the new algorithm requires indirect fetches of gradient statistics by row index, since these values are accessed in order of feature. This is a non-continuous memory access. A naive implementation of split enumeration introduces immediate read/write dependency between the accumulation and the non-continuous memory fetch operation (see Fig. 8). This slows down split finding when the gradient statistics do not fit into CPU cache and cache miss occur.

For the exact greedy algorithm, we can alleviate the problem by a cache-aware prefetching algorithm. Specifically, we allocate an internal buffer in each thread, fetch the gradient statistics into it, and then perform accumulation in a mini-batch manner. This prefetching changes the direct read/write dependency to a longer dependency and helps to reduce the runtime overhead when number of rows in the is large. Figure 7 gives the comparison of cache-aware vs. non cache-aware algorithm on the the Higgs and the Allstate dataset. We find that cache-aware implementation of the exact greedy algorithm runs twice as fast as the naive version when the dataset is large.



(a) Allstate 10M

(b) Higgs 10M

Figure 9: The impact of block size in the approximate algorithm. We find that overly small blocks results in inefficient parallelization, while overly large blocks also slows down training due to cache misses.

For approximate algorithms, we solve the problem by choosing a correct block size. We define the block size to be maximum number of examples in contained in a block, as this reflects the cache storage cost of gradient statistics. Choosing an overly small block size results in small workload for each thread and leads to inefficient parallelization. On the other hand, overly large blocks result in cache misses, as the gradient statistics do not fit into the CPU cache. A good choice of block size balances these two factors. We compared various choices of block size on two data sets. The results are given in Fig. 9. This result validates our discussion and

Table 1: Comparison of major tree boosting systems.

System	exact greedy	approximate global	approximate local	out-of-core	sparsity aware	parallel
XGBoost	yes	yes	yes	yes	yes	yes
pGBRT	no	no	yes	no	no	yes
Spark MLLib	no	yes	no	no	partially	yes
H2O	no	yes	no	no	partially	yes
scikit-learn	yes	no	no	no	no	no
R GBM	yes	no	no	no	partially	no

shows that choosing 2^{16} examples per block balances the cache property and parallelization.

4.3 Blocks for Out-of-core Computation

One goal of our system is to fully utilize a machine's resources to achieve scalable learning. Besides processors and memory, it is important to utilize disk space to handle data that does not fit into main memory. To enable out-of-core computation, we divide the data into multiple blocks and store each block on disk. During computation, it is important to use an independent thread to pre-fetch the block into a main memory buffer, so computation can happen in concurrence with disk reading. However, this does not entirely solve the problem since the disk reading takes most of the computation time. It is important to reduce the overhead and increase the throughput of disk IO. We mainly use two techniques to improve the out-of-core computation.

Block Compression The first technique we use is block compression. The block is compressed by columns, and decompressed on the fly by an independent thread when loading into main memory. This helps to trade some of the computation in decompression with the disk reading cost. We use a general purpose compression algorithm for compressing the features values. For the row index, we substract the row index by the begining index of the block and use a 16bit integer to store each offset. This requires 2^{16} examples per block, which is confirmed to be a good setting. In most of the dataset we tested, we achieve roughly a 26% to 29% compression ratio.

Block Sharding The second technique is to shard the data onto multiple disks in an alternative manner. A pre-fetcher thread is assigned to each disk and fetches the data into an in-memory buffer. The training thread then alternatively reads the data from each buffer. This helps to increase the throughput of disk reading when multiple disks are available.

5. RELATED WORKS

Our system implements gradient boosting [10], which performs additive optimization in functional space. Gradient tree boosting has been successfully used in classification [12], learning to rank [5], structured prediction [8] as well as other fields. XGBoost incorporates a regularized model to prevent overfitting. This this resembles previous work on regularized greedy forest [25], but simplifies the objective and algorithm for parallelization. Column sampling is a simple but effective technique borrowed from RandomForest [4]. While sparsity-aware learning is essential in other types of models such as linear models [9], few works on tree learning have considered this topic in a principled way. The algorithm proposed in this paper is the first unified approach to handle all kinds of sparsity patterns.

There are several existing works on parallelizing tree learning [22, 19]. Most of these algorithms fall into the approximate framework described in this paper. Notably, it is also possible to partition data by columns [23] and apply the exact greedy algorithm. This is also supported in our framework, and the techniques such as cache-aware prefecthing can be used to benefit this type of algorithm. While most existing works focus on the algorithmic aspect of parallelization, our work improves in two unexplored system directions: out-of-core computation and cache-aware learning. This gives us insights on how the system and the algorithm can be jointly optimized and provides an end-to-end system that can handle large scale problems with very limited computing resources. We also summarize the comparison between our system and existing opensource implementations in Table 1.

Quantile summary (without weights) is a classical problem in the database community [14, 24]. However, the approximate tree boosting algorithm reveals a more general problem – finding quantiles on weighted data. To the best of our knowledge, the weighted quantile sketch proposed in this paper is the first method to solve this problem. The weighted quantile summary is also not specific to the tree learning and can benefit other applications in data science and machine learning in the future.

6. END TO END EVALUATIONS

6.1 System Implementation

We implemented XGBoost as an open source package[5]. The package is portable and reusable. It supports various weighted classification and rank objective functions, as well as user defined objective function. It is available in popular languages such as python, R, Julia and integrates naturally with language native data science pipelines such as scikit-learn. The distributed version is built on top of the rabit library[6] for allreduce. The portability of XGBoost makes it available in many ecosystems, instead of only being tied to a specific platform. The distributed XGBoost runs natively on Hadoop, MPI Sun Grid engine. Recently, we also enable distributed XGBoost on jvm bigdata stacks such as Flink and Spark. The distributed version has also been integrated into cloud platform Tianchi[7] of Alibaba. We believe that there will be more integrations in the future.

6.2 Dataset and Setup

[5] https://github.com/dmlc/xgboost
[6] https://github.com/dmlc/rabit
[7] https://tianchi.aliyun.com

Table 2: Dataset used in the Experiments.

Dataset	n	m	Task
Allstate	10 M	4227	Insurance claim classification
Higgs Boson	10 M	28	Event classification
Yahoo LTRC	473K	700	Learning to Rank
Criteo	1.7 B	67	Click through rate prediction

We used four datasets in our experiments. A summary of these datasets is given in Table 2. In some of the experiments, we use a randomly selected subset of the data either due to slow baselines or to demonstrate the performance of the algorithm with varying dataset size. We use a suffix to denote the size in these cases. For example Allstate-10K means a subset of the Allstate dataset with 10K instances.

The first dataset we use is the Allstate insurance claim dataset[8]. The task is to predict the likelihood and cost of an insurance claim given different risk factors. In the experiment, we simplified the task to only predict the likelihood of an insurance claim. This dataset is used to evaluate the impact of sparsity-aware algorithm in Sec. 3.4. Most of the sparse features in this data come from one-hot encoding. We randomly select 10M instances as training set and use the rest as evaluation set.

The second dataset is the Higgs boson dataset[9] from high energy physics. The data was produced using Monte Carlo simulations of physics events. It contains 21 kinematic properties measured by the particle detectors in the accelerator. It also contains seven additional derived physics quantities of the particles. The task is to classify whether an event corresponds to the Higgs boson. We randomly select 10M instances as training set and use the rest as evaluation set.

The third dataset is the Yahoo! learning to rank challenge dataset [6], which is one of the most commonly used benchmarks in learning to rank algorithms. The dataset contains 20K web search queries, with each query corresponding to a list of around 22 documents. The task is to rank the documents according to relevance of the query. We use the official train test split in our experiment.

The last dataset is the criteo terabyte click log dataset[10]. We use this dataset to evaluate the scaling property of the system in the out-of-core and the distributed settings. The data contains 13 integer features and 26 ID features of user, item and advertiser information. Since a tree based model is better at handling continuous features, we preprocess the data by calculating the statistics of average CTR and count of ID features on the first ten days, replacing the ID features by the corresponding count statistics during the next ten days for training. The training set after preprocessing contains 1.7 billion instances with 67 features (13 integer, 26 average CTR statistics and 26 counts). The entire dataset is more than one terabyte in LibSVM format.

We use the first three datasets for the single machine parallel setting, and the last dataset for the distributed and out-of-core settings. All the single machine experiments are conducted on a Dell PowerEdge R420 with two eight-core Intel Xeon (E5-2470) (2.3GHz) and 64GB of memory. If not specified, all the experiments are run using all the available cores in the machine. The machine settings of the distributed and the out-of-core experiments will be described in the corresponding section. In all the experiments, we boost trees with a common setting of maximum depth equals 8, shrinkage equals 0.1 and no column subsampling unless explicitly specified. We can find similar results when we use other settings of maximum depth.

[8]https://www.kaggle.com/c/ClaimPredictionChallenge
[9]https://archive.ics.uci.edu/ml/datasets/HIGGS
[10]http://labs.criteo.com/downloads/download-terabyte-click-logs/

Table 3: Comparison of Exact Greedy Methods with 500 trees on Higgs-1M data.

Method	Time per Tree (sec)	Test AUC
XGBoost	0.6841	0.8304
XGBoost (colsample=0.5)	0.6401	0.8245
scikit-learn	28.51	0.8302
R.gbm	1.032	0.6224

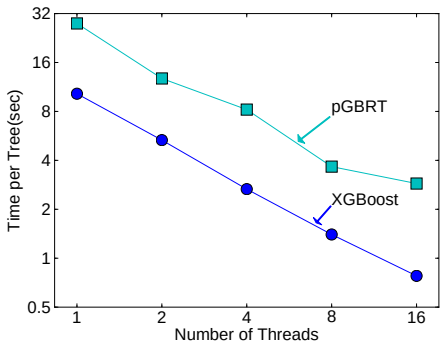


Figure 10: Comparison between XGBoost and pGBRT on Yahoo LTRC dataset.

Table 4: Comparison of Learning to Rank with 500 trees on Yahoo! LTRC Dataset

Method	Time per Tree (sec)	NDCG@10
XGBoost	0.826	0.7892
XGBoost (colsample=0.5)	0.506	0.7913
pGBRT [22]	2.576	0.7915

6.3 Classification

In this section, we evaluate the performance of XGBoost on a single machine using the exact greedy algorithm on Higgs-1M data, by comparing it against two other commonly used exact greedy tree boosting implementations. Since scikit-learn only handles non-sparse input, we choose the dense Higgs dataset for a fair comparison. We use the 1M subset to make scikit-learn finish running in reasonable time. Among the methods in comparison, R's GBM uses a greedy approach that only expands one branch of a tree, which makes it faster but can result in lower accuracy, while both scikit-learn and XGBoost learn a full tree. The results are shown in Table 3. Both XGBoost and scikit-learn give better performance than R's GBM, while XGBoost runs more than 10x faster than scikit-learn. In this experiment, we also find column subsamples gives slightly worse performance than using all the features. This could due to the fact that there are few important features in this dataset and we can benefit from greedily select from all the features.

6.4 Learning to Rank

We next evaluate the performance of XGBoost on the

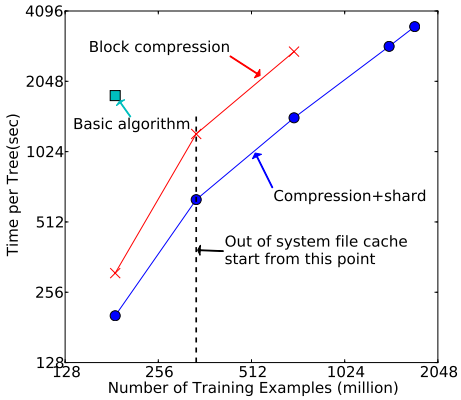


Figure 11: Comparison of out-of-core methods on different subsets of criteo data. The missing data points are due to out of disk space. We can find that basic algorithm can only handle 200M examples. Adding compression gives 3x speedup, and sharding into two disks gives another 2x speedup. The system runs out of file cache start from 400M examples. The algorithm really has to rely on disk after this point. The compression+shard method has a less dramatic slowdown when running out of file cache, and exhibits a linear trend afterwards.

learning to rank problem. We compare against pGBRT [22], the best previously pubished system on this task. XGBoost runs exact greedy algorithm, while pGBRT only support an approximate algorithm. The results are shown in Table 4 and Fig. 10. We find that XGBoost runs faster. Interestingly, subsampling columns not only reduces running time, and but also gives a bit higher performance for this problem. This could due to the fact that the subsampling helps prevent overfitting, which is observed by many of the users.

6.5 Out-of-core Experiment

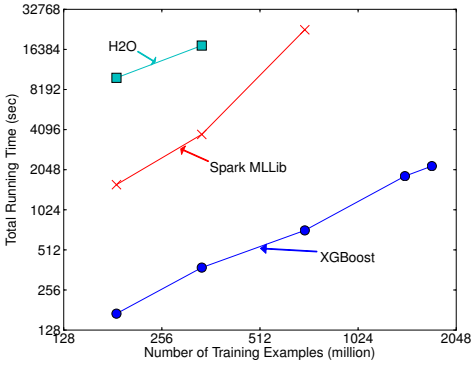
We also evaluate our system in the out-of-core setting on the criteo data. We conducted the experiment on one AWS c3.8xlarge machine (32 vcores, two 320 GB SSD, 60 GB RAM). The results are shown in Figure 11. We can find that compression helps to speed up computation by factor of three, and sharding into two disks further gives 2x speedup. For this type of experiment, it is important to use a very large dataset to drain the system file cache for a real out-of-core setting. This is indeed our setup. We can observe a transition point when the system runs out of file cache. Note that the transition in the final method is less dramatic. This is due to larger disk throughput and better utilization of computation resources. Our final method is able to process 1.7 billion examples on a single machine.

6.6 Distributed Experiment

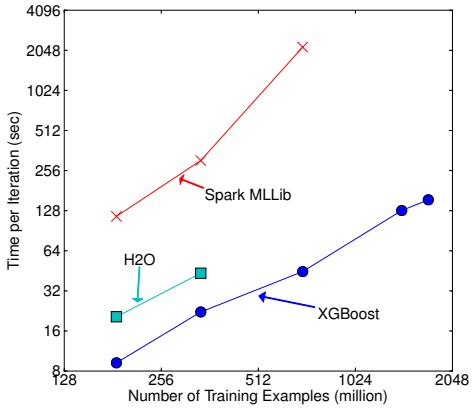
Finally, we evaluate the system in the distributed setting. We set up a YARN cluster on EC2 with m3.2xlarge machines, which is a very common choice for clusters. Each machine contains 8 virtual cores, 30GB of RAM and two 80GB SSD local disks. The dataset is stored on AWS S3 instead of HDFS to avoid purchasing persistent storage.

We first compare our system against two production-level distributed systems: Spark MLLib [18] and H2O [11]. We use

[11] www.h2o.ai



(a) End-to-end time cost include data loading



(b) Per iteration cost exclude data loading

Figure 12: Comparison of different distributed systems on 32 EC2 nodes for 10 iterations on different subset of criteo data. XGBoost runs more 10x than spark per iteration and 2.2x as H2O's optimized version (However, H2O is slow in loading the data, getting worse end-to-end time). Note that spark suffers from drastic slow down when running out of memory. XGBoost runs faster and scales smoothly to the full 1.7 billion examples with given resources by utilizing out-of-core computation.

32 m3.2xlarge machines and test the performance of the systems with various input size. Both of the baseline systems are in-memory analytics frameworks that need to store the data in RAM, while XGBoost can switch to out-of-core setting when it runs out of memory. The results are shown in Fig. 12. We can find that XGBoost runs faster than the baseline systems. More importantly, it is able to take advantage of out-of-core computing and smoothly scale to all 1.7 billion examples with the given limited computing resources. The baseline systems are only able to handle subset of the data with the given resources. This experiment shows the advantage to bring all the system improvement together and solve a real-world scale problem. We also evaluate the scaling property of XGBoost by varying the number of machines. The results are shown in Fig. 13. We can find XGBoost's performance scales linearly as we add more machines. Importantly, XGBoost is able to handle the entire 1.7 billion data with only four machines. This shows the system's potential to handle even larger data.

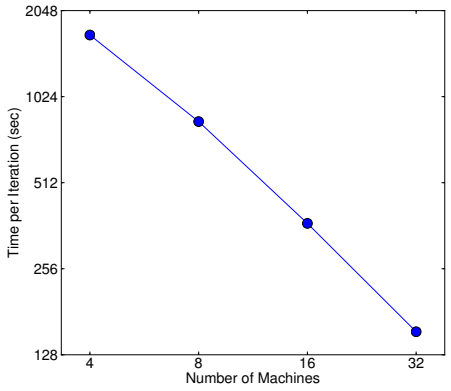


Figure 13: Scaling of XGBoost with different number of machines on criteo full 1.7 billion dataset. Using more machines results in more file cache and makes the system run faster, causing the trend to be slightly super linear. XGBoost can process the entire dataset using as little as four machines, and scales smoothly by utilizing more available resources.

7. CONCLUSION

In this paper, we described the lessons we learnt when building XGBoost, a scalable tree boosting system that is widely used by data scientists and provides state-of-the-art results on many problems. We proposed a novel sparsity aware algorithm for handling sparse data and a theoretically justified weighted quantile sketch for approximate learning. Our experience shows that cache access patterns, data compression and sharding are essential elements for building a scalable end-to-end system for tree boosting. These lessons can be applied to other machine learning systems as well. By combining these insights, XGBoost is able to solve real-world scale problems using a minimal amount of resources.

Acknowledgments

We would like to thank Tyler B. Johnson, Marco Tulio Ribeiro, Sameer Singh, Arvind Krishnamurthy for their valuable feedback. We also sincerely thank Tong He, Bing Xu, Michael Benesty, Yuan Tang, Hongliang Liu, Qiang Kou, Nan Zhu and all other contributors in the XGBoost community. This work was supported in part by ONR (PECASE) N000141010672, NSF IIS 1258741 and the TerraSwarm Research Center sponsored by MARCO and DARPA.

8. REFERENCES

[1] R. Bekkerman. The present and the future of the kdd cup competition: an outsider's perspective.

[2] R. Bekkerman, M. Bilenko, and J. Langford. *Scaling Up Machine Learning: Parallel and Distributed Approaches*. Cambridge University Press, New York, NY, USA, 2011.

[3] J. Bennett and S. Lanning. The netflix prize. In *Proceedings of the KDD Cup Workshop 2007*, pages 3–6, New York, Aug. 2007.

[4] L. Breiman. Random forests. *Maching Learning*, 45(1):5–32, Oct. 2001.

[5] C. Burges. From ranknet to lambdarank to lambdamart: An overview. *Learning*, 11:23–581, 2010.

[6] O. Chapelle and Y. Chang. Yahoo! Learning to Rank Challenge Overview. *Journal of Machine Learning Research - W & CP*, 14:1–24, 2011.

[7] T. Chen, H. Li, Q. Yang, and Y. Yu. General functional matrix factorization using gradient boosting. In *Proceeding of 30th International Conference on Machine Learning (ICML'13)*, volume 1, pages 436–444, 2013.

[8] T. Chen, S. Singh, B. Taskar, and C. Guestrin. Efficient second-order gradient boosting for conditional random fields. In *Proceeding of 18th Artificial Intelligence and Statistics Conference (AISTATS'15)*, volume 1, 2015.

[9] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[10] J. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of Statistics*, 29(5):1189–1232, 2001.

[11] J. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002.

[12] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. *Annals of Statistics*, 28(2):337–407, 2000.

[13] J. H. Friedman and B. E. Popescu. Importance sampled learning ensembles, 2003.

[14] M. Greenwald and S. Khanna. Space-efficient online computation of quantile summaries. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data*, pages 58–66, 2001.

[15] X. He, J. Pan, O. Jin, T. Xu, B. Liu, T. Xu, Y. Shi, A. Atallah, R. Herbrich, S. Bowers, and J. Q. n. Candela. Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, ADKDD'14, 2014.

[16] P. Li. Robust Logitboost and adaptive base class (ABC) Logitboost. In *Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI'10)*, pages 302–311, 2010.

[17] P. Li, Q. Wu, and C. J. Burges. Mcrank: Learning to rank using multiple classification and gradient boosting. In *Advances in Neural Information Processing Systems 20*, pages 897–904. 2008.

[18] X. Meng, J. Bradley, B. Yavuz, E. Sparks, S. Venkataraman, D. Liu, J. Freeman, D. Tsai, M. Amde, S. Owen, D. Xin, R. Xin, M. J. Franklin, R. Zadeh, M. Zaharia, and A. Talwalkar. MLlib: Machine learning in apache spark. *Journal of Machine Learning Research*, 17(34):1–7, 2016.

[19] B. Panda, J. S. Herbach, S. Basu, and R. J. Bayardo. Planet: Massively parallel learning of tree ensembles with mapreduce. *Proceeding of VLDB Endowment*, 2(2):1426–1437, Aug. 2009.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[21] G. Ridgeway. *Generalized Boosted Models: A guide to the gbm package*.

[22] S. Tyree, K. Weinberger, K. Agrawal, and J. Paykin. Parallel boosted regression trees for web search ranking. In *Proceedings of the 20th international conference on World wide web*, pages 387–396. ACM, 2011.

[23] J. Ye, J.-H. Chow, J. Chen, and Z. Zheng. Stochastic gradient boosted distributed decision trees. In *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, CIKM '09.

[24] Q. Zhang and W. Wang. A fast algorithm for approximate quantiles in high speed data streams. In *Proceedings of the 19th International Conference on Scientific and Statistical Database Management*, 2007.

[25] T. Zhang and R. Johnson. Learning nonlinear functions using regularized greedy forest. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5), 2014.

APPENDIX

A. WEIGHTED QUANTILE SKETCH

In this section, we introduce the weighted quantile sketch algorithm. Approximate answer of quantile queries is for many real-world applications. One classical approach to this problem is GK algorithm [14] and extensions based on the GK framework [24]. The main component of these algorithms is a data structure called quantile summary, that is able to answer quantile queries with relative accuracy of ϵ. Two operations are defined for a quantile summary:

- A merge operation that combines two summaries with approximation error ϵ_1 and ϵ_2 together and create a merged summary with approximation error $\max(\epsilon_1, \epsilon_2)$.
- A prune operation that reduces the number of elements in the summary to $b+1$ and changes approximation error from ϵ to $\epsilon + \frac{1}{b}$.

A quantile summary with merge and prune operations forms basic building blocks of the distributed and streaming quantile computing algorithms [24].

In order to use quantile computation for approximate tree boosting, we need to find quantiles on weighted data. This more general problem is not supported by any of the existing algorithm. In this section, we describe a non-trivial weighted quantile summary structure to solve this problem. Importantly, the new algorithm contains merge and prune operations with *the same guarantee* as GK summary. This allows our summary to be plugged into all the frameworks used GK summary as building block and answer quantile queries over weighted data efficiently.

A.1 Formalization and Definitions

Given an input multi-set $\mathcal{D} = \{(x_1, w_1), (x_2, w_2) \cdots (x_n, w_n)\}$ such that $w_i \in [0, +\infty), x_i \in \mathcal{X}$. Each x_i corresponds to a position of the point and w_i is the weight of the point. Assume we have a total order $<$ defined on $\mathcal{X}$. Let us define two rank functions $r^-_{\mathcal{D}}, r^+_{\mathcal{D}} : \mathcal{X} \to [0, +\infty)$

$$r^-_{\mathcal{D}}(y) = \sum_{(x,w)\in\mathcal{D}, x<y} w \tag{10}$$

$$r^+_{\mathcal{D}}(y) = \sum_{(x,w)\in\mathcal{D}, x\leq y} w \tag{11}$$

We should note that since $\mathcal{D}$ is defined to be a *multiset* of the points. It can contain multiple record with exactly same position x and weight w. We also define another weight function $\omega_{\mathcal{D}} : \mathcal{X} \to [0, +\infty)$ as

$$\omega_{\mathcal{D}}(y) = r^+_{\mathcal{D}}(y) - r^-_{\mathcal{D}}(y) = \sum_{(x,w)\in\mathcal{D}, x=y} w. \tag{12}$$

Finally, we also define the weight of multi-set $\mathcal{D}$ to be the sum of weights of all the points in the set

$$\omega(\mathcal{D}) = \sum_{(x,w)\in\mathcal{D}} w \tag{13}$$

Our task is given a series of input $\mathcal{D}$, to estimate $r^+(y)$ and $r^-(y)$ for $y \in \mathcal{X}$ as well as finding points with specific rank. Given these notations, we define quantile summary of weighted examples as follows:

DEFINITION A.1. *Quantile Summary of Weighted Data*
A quantile summary for $\mathcal{D}$ is defined to be tuple $Q(\mathcal{D}) = (S, \tilde{r}^+_{\mathcal{D}}, \tilde{r}^-_{\mathcal{D}}, \tilde{\omega}_{\mathcal{D}})$, where $S = \{x_1, x_2, \cdots, x_k\}$ is selected from the points in $\mathcal{D}$ (i.e. $x_i \in \{x|(x, w) \in \mathcal{D}\}$) with the following properties:

1) $x_i < x_{i+1}$ for all i, and x_1 and x_k are minimum and maximum point in $\mathcal{D}$:

$$x_1 = \min_{(x,w)\in\mathcal{D}} x, \quad x_k = \max_{(x,w)\in\mathcal{D}} x$$

2) $\tilde{r}^+_{\mathcal{D}}$, $\tilde{r}^-_{\mathcal{D}}$ and $\tilde{\omega}_{\mathcal{D}}$ are functions in $S \to [0, +\infty)$, that satisfies

$$\tilde{r}^-_{\mathcal{D}}(x_i) \leq r^-_{\mathcal{D}}(x_i), \quad \tilde{r}^+_{\mathcal{D}}(x_i) \geq r^+_{\mathcal{D}}(x_i), \quad \tilde{\omega}_{\mathcal{D}}(x_i) \leq \omega_{\mathcal{D}}(x_i), \tag{14}$$

the equality sign holds for maximum and minimum point ($\tilde{r}^-_{\mathcal{D}}(x_i) = r^-_{\mathcal{D}}(x_i)$, $\tilde{r}^+_{\mathcal{D}}(x_i) = r^+_{\mathcal{D}}(x_i)$ and $\tilde{\omega}_{\mathcal{D}}(x_i) = \omega_{\mathcal{D}}(x_i)$ for $i \in \{1, k\}$). Finally, the function value must also satisfy the following constraints

$$\tilde{r}^-_{\mathcal{D}}(x_i) + \tilde{\omega}_{\mathcal{D}}(x_i) \leq \tilde{r}^-_{\mathcal{D}}(x_{i+1}), \quad \tilde{r}^+_{\mathcal{D}}(x_i) \leq \tilde{r}^+_{\mathcal{D}}(x_{i+1}) - \tilde{\omega}_{\mathcal{D}}(x_{i+1}) \tag{15}$$

Since these functions are only defined on S, it is suffice to use $4k$ record to store the summary. Specifically, we need to remember each x_i and the corresponding function values of each x_i.

DEFINITION A.2. *Extension of Function Domains*
Given a quantile summary $Q(\mathcal{D}) = (S, \tilde{r}^+_{\mathcal{D}}, \tilde{r}^-_{\mathcal{D}}, \tilde{\omega}_{\mathcal{D}})$ defined in Definition A.1, the domain of $\tilde{r}^+_{\mathcal{D}}$, $\tilde{r}^-_{\mathcal{D}}$ and $\tilde{\omega}_{\mathcal{D}}$ were defined only in S. We extend the definition of these functions to $\mathcal{X} \to [0, +\infty)$ as follows
When $y < x_1$:

$$\tilde{r}^-_{\mathcal{D}}(y) = 0, \ \tilde{r}^+_{\mathcal{D}}(y) = 0, \ \tilde{\omega}_{\mathcal{D}}(y) = 0 \tag{16}$$

When $y > x_k$:

$$\tilde{r}^-_{\mathcal{D}}(y) = \tilde{r}^+_{\mathcal{D}}(x_k), \ \tilde{r}^+_{\mathcal{D}}(y) = \tilde{r}^+_{\mathcal{D}}(x_k), \ \tilde{\omega}_{\mathcal{D}}(y) = 0 \tag{17}$$

When $y \in (x_i, x_{i+1})$ for some i:

$$\begin{aligned} \tilde{r}^-_{\mathcal{D}}(y) &= \tilde{r}^-_{\mathcal{D}}(x_i) + \tilde{\omega}_{\mathcal{D}}(x_i), \\ \tilde{r}^+_{\mathcal{D}}(y) &= \tilde{r}^+_{\mathcal{D}}(x_{i+1}) - \tilde{\omega}_{\mathcal{D}}(x_{i+1}), \\ \tilde{\omega}_{\mathcal{D}}(y) &= 0 \end{aligned} \tag{18}$$

LEMMA A.1. *Extended Constraint*
The extended definition of $\tilde{r}^-_{\mathcal{D}}$, $\tilde{r}^+_{\mathcal{D}}$, $\tilde{\omega}_{\mathcal{D}}$ satisfies the following constraints

$$\tilde{r}^-_{\mathcal{D}}(y) \leq r^-_{\mathcal{D}}(y), \quad \tilde{r}^+_{\mathcal{D}}(y) \geq r^+_{\mathcal{D}}(y), \quad \tilde{\omega}_{\mathcal{D}}(y) \leq \omega_{\mathcal{D}}(y) \tag{19}$$

$$\tilde{r}^-_{\mathcal{D}}(y) + \tilde{\omega}_{\mathcal{D}}(y) \leq \tilde{r}^-_{\mathcal{D}}(x), \quad \tilde{r}^+_{\mathcal{D}}(y) \leq \tilde{r}^+_{\mathcal{D}}(x) - \tilde{\omega}_{\mathcal{D}}(x), \ \textit{for all } y < x \tag{20}$$

PROOF. The only non-trivial part is to prove the case when $y \in (x_i, x_{i+1})$:

$$\tilde{r}^-_{\mathcal{D}}(y) = \tilde{r}^-_{\mathcal{D}}(x_i) + \tilde{\omega}_{\mathcal{D}}(x_i) \leq r^-_{\mathcal{D}}(x_i) + \omega_{\mathcal{D}}(x_i) \leq r^-_{\mathcal{D}}(y)$$

$$\tilde{r}^+_{\mathcal{D}}(y) = \tilde{r}^+_{\mathcal{D}}(x_{i+1}) - \tilde{\omega}_{\mathcal{D}}(x_{i+1}) \geq r^+_{\mathcal{D}}(x_{i+1}) - \omega_{\mathcal{D}}(x_{i+1}) \geq r^+_{\mathcal{D}}(y)$$

This proves Eq. (19). Furthermore, we can verify that

$$\tilde{r}^+_{\mathcal{D}}(x_i) \leq \tilde{r}^+_{\mathcal{D}}(x_{i+1}) - \tilde{\omega}_{\mathcal{D}}(x_{i+1}) = \tilde{r}^+_{\mathcal{D}}(y) - \tilde{\omega}_{\mathcal{D}}(y)$$

$$\tilde{r}^-_{\mathcal{D}}(y) + \tilde{\omega}_{\mathcal{D}}(y) = \tilde{r}^-_{\mathcal{D}}(x_i) + \tilde{\omega}_{\mathcal{D}}(x_i) + 0 \leq \tilde{r}^-_{\mathcal{D}}(x_{i+1})$$

$$\tilde{r}^+_{\mathcal{D}}(y) = \tilde{r}^+_{\mathcal{D}}(x_{i+1}) - \tilde{\omega}_{\mathcal{D}}(x_{i+1})$$

Using these facts and transitivity of $<$ relation, we can prove Eq. (20) □

We should note that the extension is based on the ground case defined in S, and we do not require extra space to store the summary in order to use the extended definition. We are now ready to introduce the definition of ϵ-approximate quantile summary.

DEFINITION A.3. *ϵ-Approximate Quantile Summary*
Given a quantile summary $Q(\mathcal{D}) = (S, \tilde{r}^+_{\mathcal{D}}, \tilde{r}^-_{\mathcal{D}}, \tilde{\omega}_{\mathcal{D}})$, we call it is ϵ-approximate summary if for any $y \in \mathcal{X}$

$$\tilde{r}^+_{\mathcal{D}}(y) - \tilde{r}^-_{\mathcal{D}}(y) - \tilde{\omega}_{\mathcal{D}}(y) \leq \epsilon\omega(\mathcal{D}) \tag{21}$$

We use this definition since we know that $r^-(y) \in [\tilde{r}^-_{\mathcal{D}}(y), \tilde{r}^+_{\mathcal{D}}(y) - \tilde{\omega}_{\mathcal{D}}(y)]$ and $r^+(y) \in [\tilde{r}^-_{\mathcal{D}}(y) + \tilde{\omega}_{\mathcal{D}}(y), \tilde{r}^+_{\mathcal{D}}(y)]$. Eq. (21) means the we can get estimation of $r^+(y)$ and $r^-(y)$ by error of at most $\epsilon\omega(\mathcal{D})$.

LEMMA A.2. *Quantile summary $Q(\mathcal{D}) = (S, \tilde{r}^+_{\mathcal{D}}, \tilde{r}^-_{\mathcal{D}}, \tilde{\omega}_{\mathcal{D}})$ is an ϵ-approximate summary if and only if the following two condition holds*

$$\tilde{r}^+_{\mathcal{D}}(x_i) - \tilde{r}^-_{\mathcal{D}}(x_i) - \tilde{\omega}_{\mathcal{D}}(x_i) \leq \epsilon\omega(\mathcal{D}) \tag{22}$$

$$\tilde{r}^+_{\mathcal{D}}(x_{i+1}) - \tilde{r}^-_{\mathcal{D}}(x_i) - \tilde{\omega}_{\mathcal{D}}(x_{i+1}) - \tilde{\omega}_{\mathcal{D}}(x_i) \leq \epsilon\omega(\mathcal{D}) \tag{23}$$

PROOF. *The key is again consider $y \in (x_i, x_{i+1})$*

$$\tilde{r}^+_{\mathcal{D}}(y) - \tilde{r}^-_{\mathcal{D}}(y) - \tilde{\omega}_{\mathcal{D}}(y) = [\tilde{r}^+_{\mathcal{D}}(x_{i+1}) - \tilde{\omega}_{\mathcal{D}}(x_{i+1})] - [\tilde{r}^+_{\mathcal{D}}(x_i) + \tilde{\omega}_{\mathcal{D}}(x_i)] - 0$$

This means the condition in Eq. (23) plus Eq.(22) can give us Eq. (21) □

Property of Extended Function In this section, we have introduced the extension of function $\tilde{r}^+_{\mathcal{D}}, \tilde{r}^-_{\mathcal{D}}, \tilde{\omega}_{\mathcal{D}}$ to $\mathcal{X} \to [0, +\infty)$. The key theme discussed in this section is the relation of *constraints on the original function and constraints on the extended function*. Lemma A.1 and A.2 show that the constraints on the original function can lead to in more general constraints on the extended function. This is a very useful property which will be used in the proofs in later sections.

A.2 Construction of Initial Summary

Given a small multi-set $\mathcal{D} = \{(x_1, w_1), (x_2, w_2), \cdots, (x_n, w_n)\}$, we can construct initial summary $Q(\mathcal{D}) = \{S, \tilde{r}^+_{\mathcal{D}}, \tilde{r}^-_{\mathcal{D}}, \tilde{\omega}_{\mathcal{D}}\}$, with S to the set of all values in $\mathcal{D}$ ($S = \{x | (x, w) \in \mathcal{D}\}$), and $\tilde{r}^+_{\mathcal{D}}, \tilde{r}^-_{\mathcal{D}}, \tilde{\omega}_{\mathcal{D}}$ defined to be

$$\tilde{r}^+_{\mathcal{D}}(x) = r^+_{\mathcal{D}}(x), \quad \tilde{r}^-_{\mathcal{D}}(x) = r^-_{\mathcal{D}}(x), \quad \tilde{\omega}_{\mathcal{D}}(x) = \omega_{\mathcal{D}}(x) \text{ for } x \in S \tag{24}$$

The constructed summary is 0-approximate summary, since it can answer all the queries accurately. The constructed summary can be feed into future operations described in the latter sections.

A.3 Merge Operation

In this section, we define how we can merge the two summaries together. Assume we have $Q(\mathcal{D}_1) = (S_1, \tilde{r}^+_{\mathcal{D}_1}, \tilde{r}^-_{\mathcal{D}_1}, \tilde{\omega}_{\mathcal{D}_1})$ and $Q(\mathcal{D}_2) = (S_2, \tilde{r}^+_{\mathcal{D}_1}, \tilde{r}^-_{\mathcal{D}_2}, \tilde{\omega}_{\mathcal{D}_2})$ quantile summary of two dataset $\mathcal{D}_1$ and $\mathcal{D}_2$. Let $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$, and define the merged summary $Q(\mathcal{D}) = (S, \tilde{r}^+_{\mathcal{D}}, \tilde{r}^-_{\mathcal{D}}, \tilde{\omega}_{\mathcal{D}})$ as follows.

$$S = \{x_1, x_2 \cdots, x_k\}, x_i \in S_1 \text{ or } x_i \in S_2 \tag{25}$$

The points in S are combination of points in S_1 and S_2. And the function $\tilde{r}^+_{\mathcal{D}}, \tilde{r}^-_{\mathcal{D}}, \tilde{\omega}_{\mathcal{D}}$ are defined to be

$$\tilde{r}^-_{\mathcal{D}}(x_i) = \tilde{r}^-_{\mathcal{D}_1}(x_i) + \tilde{r}^-_{\mathcal{D}_2}(x_i) \tag{26}$$

$$\tilde{r}^+_{\mathcal{D}}(x_i) = \tilde{r}^+_{\mathcal{D}_1}(x_i) + \tilde{r}^+_{\mathcal{D}_2}(x_i) \tag{27}$$

$$\tilde{\omega}_{\mathcal{D}}(x_i) = \tilde{\omega}_{\mathcal{D}_1}(x_i) + \tilde{\omega}_{\mathcal{D}_2}(x_i) \tag{28}$$

Here we use functions defined on $S \to [0, +\infty)$ on the left sides of equalities and use the extended function definitions on the right sides.

Due to additive nature of r^+, r^- and ω, which can be formally written as

$$\begin{aligned} r^-_{\mathcal{D}}(y) =& r^-_{\mathcal{D}_1}(y) + r^-_{\mathcal{D}_2}(y), \\ r^+_{\mathcal{D}}(y) =& r^+_{\mathcal{D}_1}(y) + r^+_{\mathcal{D}_2}(y), \\ \omega_{\mathcal{D}}(y) =& \omega_{\mathcal{D}_1}(y) + \omega_{\mathcal{D}_2}(y), \end{aligned} \tag{29}$$

and the extended constraint property in Lemma A.1, we can verify that $Q(\mathcal{D})$ satisfies all the constraints in Definition A.1. Therefore it is a valid quantile summary.

LEMMA A.3. *The combined quantile summary satisfies*

$$\tilde{r}^-_{\mathcal{D}}(y) = \tilde{r}^-_{\mathcal{D}_1}(y) + \tilde{r}^-_{\mathcal{D}_2}(y) \tag{30}$$

$$\tilde{r}^+_{\mathcal{D}}(y) = \tilde{r}^+_{\mathcal{D}_1}(y) + \tilde{r}^+_{\mathcal{D}_2}(y) \tag{31}$$

$$\tilde{\omega}_{\mathcal{D}}(y) = \tilde{\omega}_{\mathcal{D}_1}(y) + \tilde{\omega}_{\mathcal{D}_2}(y) \tag{32}$$

for all $y \in \mathcal{X}$

Algorithm 4: Query Function $g(Q, d)$

Input: d: $0 \leq d \leq \omega(\mathcal{D})$
Input: $Q(\mathcal{D}) = (S, \tilde{r}^+_{\mathcal{D}}, \tilde{r}^-_{\mathcal{D}}, \tilde{\omega}_{\mathcal{D}})$ where $S = x_1, x_2, \cdots, x_k$
if $d < \frac{1}{2}[\tilde{r}^-_{\mathcal{D}}(x_1) + \tilde{r}^+_{\mathcal{D}}(x_1)]$ **then return** x_1 ;
if $d \geq \frac{1}{2}[\tilde{r}^-_{\mathcal{D}}(x_k) + \tilde{r}^+_{\mathcal{D}}(x_k)]$ **then return** x_k ;
Find i such that
$\frac{1}{2}[\tilde{r}^-_{\mathcal{D}}(x_i) + \tilde{r}^+_{\mathcal{D}}(x_i)] \leq d < \frac{1}{2}[\tilde{r}^-_{\mathcal{D}}(x_{i+1}) + \tilde{r}^+_{\mathcal{D}}(x_{i+1})]$
if $2d < \tilde{r}^-_{\mathcal{D}}(x_i) + \tilde{\omega}_{\mathcal{D}}(x_i) + \tilde{r}^+_{\mathcal{D}}(x_{i+1}) - \tilde{\omega}_{\mathcal{D}}(x_{i+1})$ **then**
  return x_i
else
  return x_{i+1}
end

This can be obtained by straight-forward application of Definition A.2.

THEOREM A.1. *If $Q(\mathcal{D}_1)$ is ϵ_1-approximate summary, and $Q(\mathcal{D}_2)$ is ϵ_2-approximate summary. Then the merged summary $Q(\mathcal{D})$ is $\max(\epsilon_1, \epsilon_2)$-approximate summary.*

PROOF. For any $y \in \mathcal{X}$, we have

$$\begin{aligned} &\tilde{r}^+_{\mathcal{D}}(y) - \tilde{r}^-_{\mathcal{D}}(y) - \tilde{\omega}_{\mathcal{D}}(y) \\ =&[\tilde{r}^+_{\mathcal{D}_1}(y) + \tilde{r}^+_{\mathcal{D}_2}(y)] - [\tilde{r}^-_{\mathcal{D}_1}(y) + \tilde{r}^-_{\mathcal{D}_2}(y)] - [\tilde{\omega}_{\mathcal{D}_1}(y) + \tilde{\omega}_{\mathcal{D}_2}(y)] \\ \leq&\epsilon_1\omega(\mathcal{D}_1) + \epsilon_2\omega(\mathcal{D}_2) \leq \max(\epsilon_1, \epsilon_2)\omega(\mathcal{D}_1 \cup \mathcal{D}_2) \end{aligned}$$

Here the first inequality is due to Lemma A.3. □

A.4 Prune Operation

Before we start discussing the prune operation, we first introduce a query function $g(Q, d)$. The definition of function is shown in Algorithm 4. For a given rank d, the function returns a x whose rank is close to d. This property is formally described in the following Lemma.

LEMMA A.4. *For a given ϵ-approximate summary $Q(\mathcal{D}) = (S, \tilde{r}^+_{\mathcal{D}}, \tilde{r}^-_{\mathcal{D}}, \tilde{\omega}_{\mathcal{D}})$, $x^* = g(Q, d)$ satisfies the following property*

$$\begin{aligned} d &\geq \tilde{r}^+_{\mathcal{D}}(x^*) - \tilde{\omega}_{\mathcal{D}}(x^*) - \frac{\epsilon}{2}\omega(\mathcal{D}) \\ d &\leq \tilde{r}^-_{\mathcal{D}}(x^*) + \tilde{\omega}_{\mathcal{D}}(x^*) + \frac{\epsilon}{2}\omega(\mathcal{D}) \end{aligned} \tag{33}$$

PROOF. We need to discuss four possible cases

- $d < \frac{1}{2}[\tilde{r}^-_{\mathcal{D}}(x_1) + \tilde{r}^+_{\mathcal{D}}(x_1)]$ and $x^* = x_1$. Note that the rank information for x_1 is accurate ($\tilde{\omega}_{\mathcal{D}}(x_1) = \tilde{r}^+_{\mathcal{D}}(x_1) = \omega(x_1)$, $\tilde{r}^-_{\mathcal{D}}(x_1) = 0$), we have

$$\begin{aligned} d &\geq 0 - \frac{\epsilon}{2}\omega(\mathcal{D}) = \tilde{r}^+_{\mathcal{D}}(x_1) - \tilde{\omega}_{\mathcal{D}}(x_1) - \frac{\epsilon}{2}\omega(\mathcal{D}) \\ d &< \frac{1}{2}[\tilde{r}^-_{\mathcal{D}}(x_1) + \tilde{r}^+_{\mathcal{D}}(x_1)] \\ &\leq \tilde{r}^-_{\mathcal{D}}(x_1) + \tilde{r}^+_{\mathcal{D}}(x_1) \\ &= \tilde{r}^-_{\mathcal{D}}(x_1) + \tilde{\omega}^+_{\mathcal{D}}(x_1) \end{aligned}$$

- $d \geq \frac{1}{2}[\tilde{r}^-_{\mathcal{D}}(x_k) + \tilde{r}^+_{\mathcal{D}}(x_k)]$ and $x^* = x_k$, then

$$\begin{aligned} d &\geq \frac{1}{2}[\tilde{r}^-_{\mathcal{D}}(x_k) + \tilde{r}^+_{\mathcal{D}}(x_k)] \\ &= \tilde{r}^+_{\mathcal{D}}(x_k) - \frac{1}{2}[\tilde{r}^+_{\mathcal{D}}(x_k) - \tilde{r}^-_{\mathcal{D}}(x_k)] \\ &= \tilde{r}^+_{\mathcal{D}}(x_k) - \frac{1}{2}\tilde{\omega}_{\mathcal{D}}(x_k) \\ d &< \omega(\mathcal{D}) + \frac{\epsilon}{2}\omega(\mathcal{D}) = \tilde{r}^-_{\mathcal{D}}(x_k) + \tilde{\omega}_{\mathcal{D}}(x_k) + \frac{\epsilon}{2}\omega(\mathcal{D}) \end{aligned}$$

- $x^* = x_i$ in the general case, then

$$\begin{aligned}
2d &< \tilde{r}^-_{\mathcal{D}}(x_i) + \tilde{\omega}_{\mathcal{D}}(x_i) + \tilde{r}^+_{\mathcal{D}}(x_{i+1}) - \tilde{\omega}_{\mathcal{D}}(x_{i+1}) \\
&= 2[\tilde{r}^-_{\mathcal{D}}(x_i) + \tilde{\omega}_{\mathcal{D}}(x_i)] + [\tilde{r}^+_{\mathcal{D}}(x_{i+1}) - \tilde{\omega}_{\mathcal{D}}(x_{i+1}) - \tilde{r}^-_{\mathcal{D}}(x_i) - \tilde{\omega}_{\mathcal{D}}(x_i)] \\
&\leq 2[\tilde{r}^-_{\mathcal{D}}(x_i) + \tilde{\omega}_{\mathcal{D}}(x_i)] + \epsilon\omega(\mathcal{D}) \\
2d &\geq \tilde{r}^-_{\mathcal{D}}(x_i) + \tilde{r}^+_{\mathcal{D}}(x_i) \\
&= 2[\tilde{r}^+_{\mathcal{D}}(x_i) - \tilde{\omega}_{\mathcal{D}}(x_i)] - [\tilde{r}^+_{\mathcal{D}}(x_i) - \tilde{\omega}_{\mathcal{D}}(x_i) - \tilde{r}^-_{\mathcal{D}}(x_i)] + \tilde{\omega}_{\mathcal{D}}(x_i) \\
&\geq 2[\tilde{r}^+_{\mathcal{D}}(x_i) - \tilde{\omega}_{\mathcal{D}}(x_i)] - \epsilon\omega(\mathcal{D}) + 0
\end{aligned}$$

- $x^* = x_{i+1}$ in the general case

$$\begin{aligned}
2d &\geq \tilde{r}^-_{\mathcal{D}}(x_i) + \tilde{\omega}_{\mathcal{D}}(x_i) + \tilde{r}^+_{\mathcal{D}}(x_{i+1}) - \tilde{\omega}_{\mathcal{D}}(x_{i+1}) \\
&= 2[\tilde{r}^+_{\mathcal{D}}(x_{i+1}) - \tilde{\omega}_{\mathcal{D}}(x_{i+1})] \\
&\quad - [\tilde{r}^+_{\mathcal{D}}(x_{i+1}) - \tilde{\omega}_{\mathcal{D}}(x_{i+1}) - \tilde{r}^-_{\mathcal{D}}(x_i) - \tilde{\omega}_{\mathcal{D}}(x_i)] \\
&\geq 2[\tilde{r}^+_{\mathcal{D}}(x_{i+1}) + \tilde{\omega}_{\mathcal{D}}(x_{i+1})] - \epsilon\omega(\mathcal{D}) \\
2d &\leq \tilde{r}^-_{\mathcal{D}}(x_{i+1}) + \tilde{r}^+_{\mathcal{D}}(x_{i+1}) \\
&= 2[\tilde{r}^-_{\mathcal{D}}(x_{i+1}) + \tilde{\omega}_{\mathcal{D}}(x_{i+1})] \\
&\quad + [\tilde{r}^+_{\mathcal{D}}(x_{i+1}) - \tilde{\omega}_{\mathcal{D}}(x_{i+1}) - \tilde{r}^-_{\mathcal{D}}(x_{i+1})] - \tilde{\omega}_{\mathcal{D}}(x_{i+1}) \\
&\leq 2[\tilde{r}^-_{\mathcal{D}}(x_{i+1}) + \tilde{\omega}_{\mathcal{D}}(x_{i+1})] + \epsilon\omega(\mathcal{D}) - 0
\end{aligned}$$

□

Now we are ready to introduce the prune operation. Given a quantile summary $Q(\mathcal{D}) = (S, \tilde{r}^+_{\mathcal{D}}, \tilde{r}^-_{\mathcal{D}}, \tilde{\omega}_{\mathcal{D}})$ with $S = \{x_1, x_2, \cdots, x_k\}$ elements, and a memory budget b. The prune operation creates another summary $Q'(\mathcal{D}) = (S', \tilde{r}^+_{\mathcal{D}}, \tilde{r}^-_{\mathcal{D}}, \tilde{\omega}_{\mathcal{D}})$ with $S' = \{x'_1, x'_2, \cdots, x'_{b+1}\}$, where x'_i are selected by query the original summary such that

$$x'_i = g\left(Q, \frac{i-1}{b}\omega(\mathcal{D})\right).$$

The definition of $\tilde{r}^+_{\mathcal{D}}, \tilde{r}^-_{\mathcal{D}}, \tilde{\omega}_{\mathcal{D}}$ in Q' is copied from original summary Q, by restricting input domain from S to S'. There could be duplicated entries in the S'. These duplicated entries can be safely removed to further reduce the memory cost. Since all the elements in Q' comes from Q, we can verify that Q' satisfies all the constraints in Definition A.1 and is a valid quantile summary.

THEOREM A.2. *Let $Q'(\mathcal{D})$ be the summary pruned from an ϵ-approximate quantile summary $Q(\mathcal{D})$ with b memory budget. Then $Q'(\mathcal{D})$ is a $(\epsilon + \frac{1}{b})$-approximate summary.*

PROOF. We only need to prove the property in Eq. (23) for Q'. Using Lemma A.4, we have

$$\begin{aligned}
\frac{i-1}{b}\omega(\mathcal{D}) + \frac{\epsilon}{2}\omega(\mathcal{D}) &\geq \tilde{r}^+_{\mathcal{D}}(x'_i) - \tilde{\omega}_{\mathcal{D}}(x'_i) \\
\frac{i-1}{b}\omega(\mathcal{D}) - \frac{\epsilon}{2}\omega(\mathcal{D}) &\leq \tilde{r}^-_{\mathcal{D}}(x'_i) + \tilde{\omega}_{\mathcal{D}}(x'_i)
\end{aligned}$$

Combining these inequalities gives

$$\begin{aligned}
&\tilde{r}^+_{\mathcal{D}}(x'_{i+1}) - \tilde{\omega}_{\mathcal{D}}(x'_{i+1}) - \tilde{r}^-_{\mathcal{D}}(x'_i) - \tilde{\omega}_{\mathcal{D}}(x'_i) \\
\leq &[\frac{i}{b}\omega(\mathcal{D}) + \frac{\epsilon}{2}\omega(\mathcal{D})] - [\frac{i-1}{b}\omega(\mathcal{D}) - \frac{\epsilon}{2}\omega(\mathcal{D})] = (\frac{1}{b} + \epsilon)\omega(\mathcal{D})
\end{aligned}$$

□

MATERIALS SCIENCE

10.1126/sciadv.1501492

Experimental test of Landauer's principle in single-bit operations on nanomagnetic memory bits

Jeongmin Hong,[1] Brian Lambson,[2] Scott Dhuey,[3] Jeffrey Bokor[1]*

Minimizing energy dissipation has emerged as the key challenge in continuing to scale the performance of digital computers. The question of whether there exists a fundamental lower limit to the energy required for digital operations is therefore of great interest. A well-known theoretical result put forward by Landauer states that any irreversible single-bit operation on a physical memory element in contact with a heat bath at a temperature T requires at least $k_BT\ln(2)$ of heat be dissipated from the memory into the environment, where k_B is the Boltzmann constant. We report an experimental investigation of the intrinsic energy loss of an adiabatic single-bit reset operation using nanoscale magnetic memory bits, by far the most ubiquitous digital storage technology in use today. Through sensitive, high-precision magnetometry measurements, we observed that the amount of dissipated energy in this process is consistent (within 2 SDs of experimental uncertainty) with the Landauer limit. This result reinforces the connection between "information thermodynamics" and physical systems and also provides a foundation for the development of practical information processing technologies that approach the fundamental limit of energy dissipation. The significance of the result includes insightful direction for future development of information technology.

INTRODUCTION

In 1961, Landauer (*1*) proposed the principle that logical irreversibility is associated with physical irreversibility and further theorized that the erasure of information is fundamentally a dissipative process. Among several seminal results, his theory states that for any irreversible single-bit operation on a physical memory element in contact with a heat bath at a given temperature, at least $k_BT\ln(2)$ of heat must be dissipated from the memory into the environment, where k_B is the Boltzmann constant and T is temperature (*2*). The single-bit reset operation process is schematically shown in Fig. 1A. As shown by Landauer (*1*, *2*), the extracted work from the process, regardless of the initial state of the bit, is $W_{operation} \geq k_BT\ln(2)$. This energy, $k_BT\ln(2)$, corresponds to a value of 2.8 zJ (2.8×10^{-21} J) at 300 K. In the field of ultra-low-energy electronics, computations that approach this energy limit are of considerable practical interest (*3*).

The first direct experimental test of Landauer's principle was reported in 2012 using a 2-μm glass bead in water manipulated in a double-well laser trap as a model system (*4*), and a higher precision measurement using 200-nm fluorescent particles in an electrokinetic feedback trap was recently reported (*5*). Although the topic is of great importance for information processing, the Landauer limit in single-bit operations has yet to be tested in any other physical system (*5*, *6*), particularly one that is relevant in practical digital devices. Therefore, confirming the generality of Landauer's principle in another, very different physical system is of great interest. Landauer and Bennett (*1*, *7*) both used nanomagnets as prototypical bistable elements in which the energy efficiency near the fundamental limits was considered. Accordingly, we report here an experimental study of Landauer's principle directly in nanomagnets.

[1]Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA 94720, USA. [2]Haynes and Boone LLP, 525 University Avenue, Palo Alto, CA 94301, USA. [3]The Molecular Foundry, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Berkeley, CA 94720, USA.
*Corresponding author. E-mail: jbokor@eecs.berkeley.edu

The fact that mesoscopic single-domain magnetic dots comprising more than 10^4 individual spins can nevertheless behave as a simple system with a single informational degree of freedom has been explicitly analyzed and confirmed theoretically and experimentally (*8*, *9*). Further theoretical studies (*10*, *11*) in which the adiabatic "reset to one" sequence for a nanomagnet memory suggested by Bennett (*7*) was explicitly simulated using the stochastic Landau-Lifschitz-Gilbert formalism, confirmed Landauer's limit of energy dissipation of $k_BT\ln(2)$ with high accuracy. For a nanomagnetic memory bit, magnetic anisotropy is used to create an "easy axis" along which the net magnetization aligns to minimize magnetostatic energy. As shown in Fig. 1A, the magnetization can align either "up" or "down" along the easy axis to represent binary "0" and "1." We denote the easy axis as the y axis. The orthogonal x axis is referred to as the "hard axis." The anisotropy of the magnet creates an energy barrier for the magnetization to align along the hard axis, allowing the nanomagnet to retain its state in the presence of thermal noise. To reset a bit stored in the nanomagnet, magnetic fields along both the x and y axes are used. The x axis field is used to lower the energy barrier between the two states, and the y axis field is then used to drive the nanomagnet into the 1 state.

RESULTS

In the micromagnetic simulations of Lambson and Madami (*10*, *11*), and as shown in Fig. 1B, the reset sequence can be divided into four steps. Initially, the nanomagnet is in either 0 or 1 state, and afterward, it is reset to the 1 state. The internal energy dissipation in the nanomagnet is found by integrating the area of m-H loops for magnetic fields applied along both the x and y axes (hard and easy axes, respectively) followed by their subtraction. To perform the hysteresis loop measurements of interest, the external magnetic fields are specified as a function of time in a quasistatic manner as illustrated in Fig. 1B.

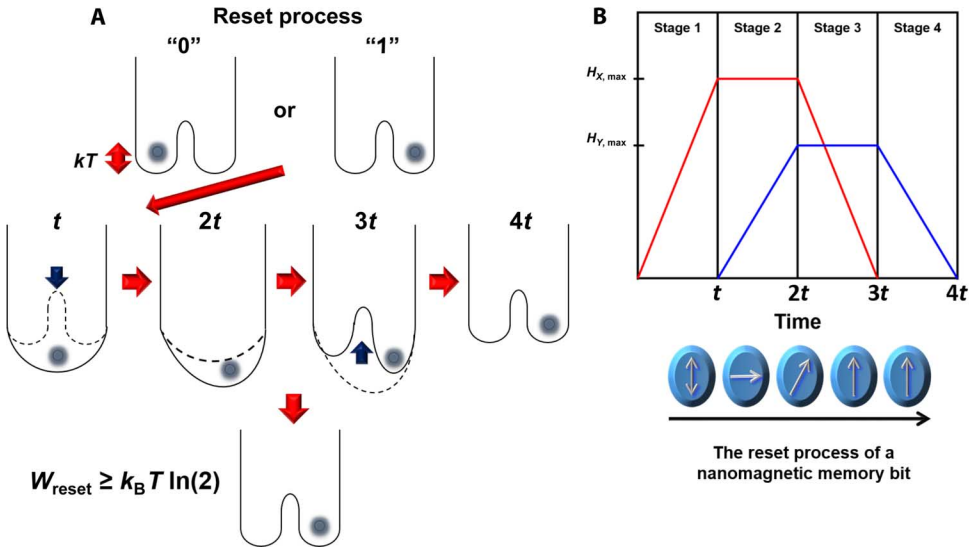


Fig. 1. Thermodynamics background. (**A**) Description of single-bit reset by time sequence. Before the erasure, the memory stores information in state 0 or 1; after the reset, the memory stores information in state 0 in accordance with the unit probability. (**B**) Timing diagram for the external magnetic fields applied during the restore-to-one process. H_x is applied along the magnetic hard axis to remove the uniaxial anisotropy barrier, whereas H_y is applied along the easy axis to force the magnetization into the 1 state. Illustrations are provided of the magnetization of the nanomagnet at the beginning and end of each stage and of the direction of the applied field in the *x*-*y* plane.

Applying the fields in this manner splits the operation into four stages, and during any given stage, one of the fields is held fixed while the other increased linearly from zero to its maximum value or vice versa, as shown in Fig. 1B. In stage 1, H_x is applied to saturate the hard axis, which removes the energy barrier and ensures that the energy dissipation is independent of the barrier height.

As explained by Bennett (*7*), whether the Landauer erasure operation is classified as reversible or irreversible depends on whether the initial state of the nanomagnet is truly unknown (that is, randomized) or known. However, in both the reversible and irreversible cases, the amount of energy transfer that occurs during the operation is $k_B T \ln(2)$. The distinction between reversible and irreversible lies in whether or not the operation can be undone by applying the fields depicted in Fig. 1B in reverse. A more complete discussion is contained in Bennett's work (*7*). Accordingly, for experimental purposes, there is no need to randomize or otherwise specially prepare the initial state of the nanomagnets to observe the $k_B T \ln(2)$ limit. This can be further justified by observing that the first stage of the reset operation depicted in Fig. 1B (applying a field along the *x* axis) is symmetric with respect to the initial orientation of the nanomagnets along the *y* axis. After the first stage, there is no remaining *y* axis component of the magnetization of the nanomagnets, so subsequent stages of the operation are independent of the initial orientation of the nanomagnets along the *y* axis. As a result, the amount of energy dissipated during the Landauer erasure operation does not depend on the initial state of the nanomagnet.

Magneto-optic Kerr effect (MOKE) in the longitudinal geometry was used to measure the in-plane magnetic moment, *m*, of a large array of identical Permalloy nanomagnets, whereas the magnetic field, *H*, was applied using a two-axis vector electromagnet. The experimental setup is shown in Fig. 2A. The lateral dimensions of the nanomagnets were less than 100 nm to ensure they were of single domain, whereas the spacing between magnets was 400 nm to avoid dipolar interactions between magnets yet provide sufficient MOKE signal. Scanning electron microscopy (SEM) images of the sample are shown in Fig. 2B. Magnetic force microscopy (MFM) was used to confirm that the nanomagnets have a single-domain structure and have sufficient anisotropy to retain state at room temperature, as shown in Fig. 2C. Longitudinal MOKE is sensitive to magnetization along only one in-plane direction (*9*), so the sample was mounted on a rotation stage, and separate measurements were made with the sample oriented to measure *m* along each of the easy and hard axes of the nanomagnets. For each measurement along the two orientations, the magnetic field along the axis of MOKE sensitivity was slowly (time scale of many seconds) ramped between positive and negative values, whereas the transverse magnetic field (perpendicular to the axis of MOKE sensitivity) was held at fixed values. The values of the transverse magnetic field were selected to generate *m*-*H* curves corresponding to each of the four steps of the reset protocol shown in Fig. 1B. The comprehensive hysteresis loops during the complete erasure process are illustrated schematically in video S1.

To quantitatively determine the net energy dissipation during the reset operation from the MOKE data, it is necessary to calibrate both the applied magnetic field and the absolute magnetization of the nanomagnets. The applied field was measured using a three-axis Hall probe sensor. To calibrate the MOKE signal, the total moment, $M_S V_T$, for the full sample was measured using a vibrating sample magnetometer (VSM). M_S is the saturation magnetization for the full sample and V_T is the total volume of the magnetic layer on the sample. An example of experimental results from one run is shown in Fig. 3. The volume of each nanomagnet, *V*, and the number of nanomagnets on the substrate were measured and calibrated using SEM for the lateral dimensions and count, and atomic force microscopy (AFM) was used to determine the thickness (see the Supplementary Materials for details). In this way, the $M_S V$ value for an individual nanomagnet from the MOKE data could be absolutely determined.

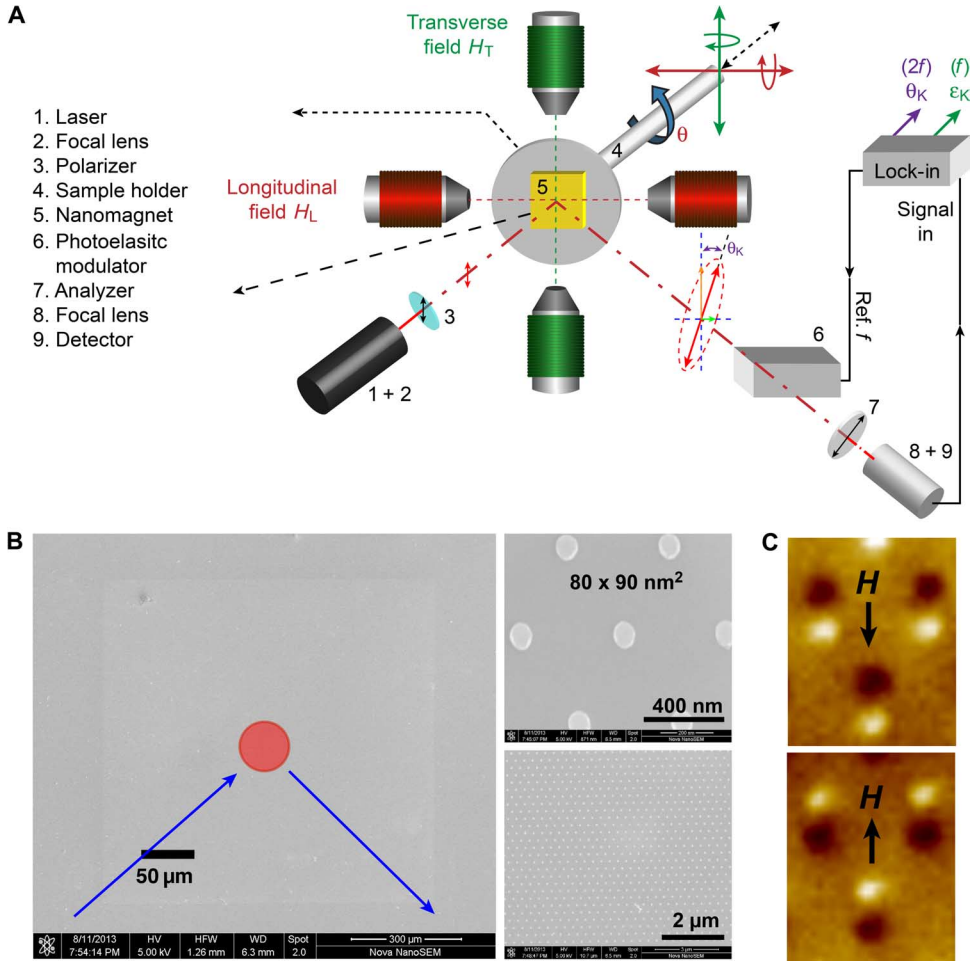


Fig. 2. The magneto-optic Kerr microscopy experimental set up. (**A**) Schematic of the experimental MOKE setup. (**B**) SEM images of the sample. The circle represents the approximate size of the probe laser spot. (**C**) MFM images of individual single-domain nanomagnets.

The energy dissipation (the magnetization energy transferred by the applied magnetic field to a nanomagnet) is determined by the total area of the hysteresis curves. As seen in Fig. 3 and video S1, the *x* and *y* hysteresis curves are traversed in opposite directions during the course of the reset operation so that the total energy is found by subtracting the area of the *y* hysteresis (easy axis) loop from the area of the *x* hysteresis (hard axis) loop. Further details concerning the calculation of energy dissipation are given in the Supplementary Materials.

As expected, the temperature dependence between 300 and 400 K was very similar to the theoretical prediction based on the average of all 2000 simulations. The mean energy dissipation was found to be 0.6842 k_BT, which corresponds to a 95% confidence interval of 0.6740 to 0.6943 k_BT. These values are in very close agreement with the Landauer limit, k_BT ln(2). The experimental energy dissipation was measured at temperatures varying from 300 to 400 K. As seen in fig. S1, the hysteresis loops for both axes individually show a clear systematic temperature dependence that is consistent with micromagnetic simulations (*10*, *11*), but the temperature dependence of the measured net energy dissipation was smaller than the run-to-run variation, as seen in Fig. 4A. For the set of runs, the energy dissipation was measured to be (4.2 ± 0.9) zJ, which corresponds to a value of (1.0 ± 0.22)k_BT (for T = 300 K). Statistical experimental results for energy dissipation are shown in Fig. 4B. As explicitly demonstrated by Lambson and Madami (*10*, *11*), for "ideal" nanomagnets, the dissipation should average exactly to k_BT ln(2) with a small run-to-run variation caused by thermal fluctuations. The average dissipation for five trials at room temperature was measured to be (6.09 ± 1.43) zJ at T = 300 K. This is consistent with a value of (1.45 ± 0.35)k_BT, which is extremely close to the Landauer limit of k_BT ln(2) or 0.69 k_BT. The quoted error was determined by combining in quadrature the uncertainties in each of the measured variables: nanomagnet area and thickness, magnetic field calibration, magnetic moment, lithographic variation, and the statistical variation among the five trials.

A small remanent magnetization was typically observed in the hard axis (*x*) direction (curves in Fig. 3 do not pass through the origin). We attribute this to fabrication variations among the nanomagnets. When the symmetry axes of the individual nanomagnets are not perfectly aligned with the axes of the applied magnetic fields, each of their remanent easy axis magnetizations will have a small component along the hard axis direction. Because of fabrication variations, there will be a distribution of misalignments. Experiments involving small rotations of the sample to find the net symmetry axis and measure the effect of a net tilt of the array with respect to the magnetic field axes are described in the Supplementary Materials. On the basis of these

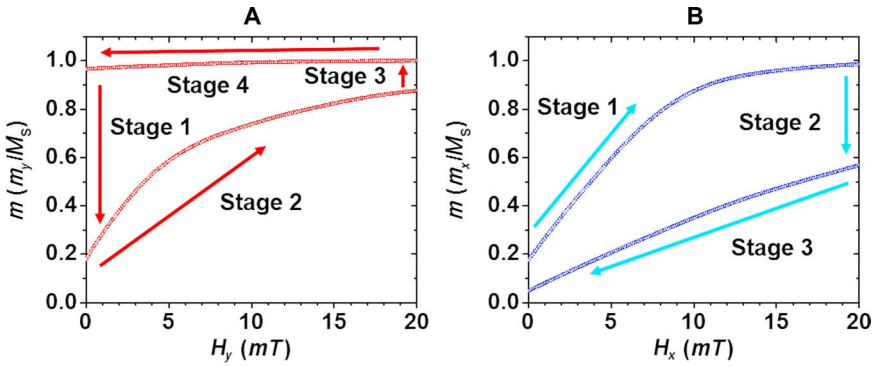
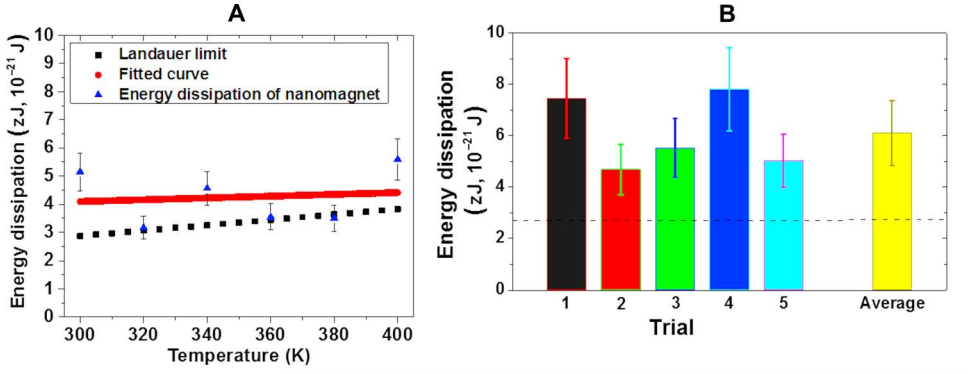


Fig. 3. The experimental *m-H* hysteresis loops of nanomagnets during the reset operation. (**A** and **B**) The m_y-H_y loop (easy axis) (A) and the m_x-H_x loop (hard axis) (B). The indicated stages correspond to the timing diagram shown in Fig. 1B.



Error	Area	Thickness	H_{calib}	M_S	Lithography	Trials	Total
%	1.2	7.9	1	10	8.2	14.3	20.8

Fig. 4. Experimental results for total energy dissipation. (**A**) The temperature dependence of energy dissipation during single-bit reset. Triangles represent experimental data obtained from integrating and subtracting hysteresis loops similar to the example shown in Fig. 3. The red line is the best fit to the experimental data. The black squares represent the Landauer limit, $k_BT\ln(2)$. (**B**) The experimentally determined energy dissipation during the reset operation. Different bars from 1 to 5 represent separate experimental runs to measure energy dissipation. The values in the table indicate estimated relative SD of the measurements of average dot area (Area), average dot thickness (Thickness), applied magnetic field (H_{calib}), saturation magnetization (M_S), residual remanence due to "tilt" effect (Lithography), and the run-to-run variation (Trials), respectively. The total experimental error was determined from the root-mean-square value for all of the variables in the table. The dotted line represents the Landauer limit, $k_BT\ln(2)$ for $T = 300$ K.

experiments, we estimate the magnitude of random variation of the symmetry axes of the nanomagnets across the array to be approximately ±1°, which is roughly consistent with the observed remanence (see also simulation in fig. S4C). We estimate that the small excess energy dissipation above the Landauer value that we observed in our experiment can be mainly attributed to this effect (see discussion in the Supplementary Materials). Other possible sources of excess dissipation include domain motion and pinning by defects and edge roughness (*12*, *13*) and edge effects. The effects of dot shape and subdomain structure on energy dissipation are explicitly considered in the simulations reported by Madami *et al.* (*11*). At this time, it is not feasible to separately estimate the magnitude of these various contributions to the small excess energy dissipation observed in our experiments. However, the fact that our results depart by only 50 to 100% of the Landauer value strongly indicates that these Permalloy nanomagnets behave very closely to the ideal "single-spin" magnets and therefore provide a practical and viable system for further exploration of the ultimate energy dissipation in information processing. As an example, a nanomagnetic logic gate for demonstrating reversible logic operation with dissipation below the Landauer limit was analyzed theoretically by Lambson *et al.* (*10*). Our results suggest that such an experiment is indeed feasible.

DISCUSSION

We have experimentally measured for the first time the intrinsic minimum energy dissipation during a single-bit operation using a nanoscale digital magnetic memory bit. Our result is within 2 SDs of the value of $k_BT\ln(2)$ predicted by Landauer. Although experimental tests of Landauer's limit have previously been performed using trapped microbeads, our result using a completely different physical system confirms its generality and, in particular, its applicability to practical information processing systems. Any practical nanomagnetic memory or logic device will inevitably involve additional energy loss

associated with the actuation mechanism (that is, the external applied magnetic fields in this experiment); however, our results demonstrate the potential to approach Landauer's limit in future information processing systems. Therefore, the significance of this result is that today's computers are far from the fundamental limit and that future marked reductions in power consumption are possible with further development of nanomagnetic memory and logic devices. Given that power consumption is the key issue that limits the continued improvement in digital computers, the result has profound suggestions for the future development of information technology.

MATERIALS AND METHODS

Fabrication of nanomagnetic memory bits

We fabricated an array of identical, single-domain, noninteracting, elliptically shaped nanomagnets by lift-off patterning of an electron beam (e-beam) evaporated amorphous Permalloy (NiFe) film on a silicon substrate using e-beam lithography. The base pressure of the e-beam evaporator was 2×10^{-7} torr, and the film thickness was 10 nm. The sample was fabricated at The Molecular Foundry at Lawrence Berkeley National Laboratory.

Dimensional metrology of nanomagnet islands using SEM and scanning probe microscopy

SEM images were collected with a Carl Zeiss LEO 1550. The statistical errors of area of the magnet and thickness measurements were 1.2 and 7.9%, respectively. The image analysis was performed using ImageJ software from the National Institutes of Health. The average size of magnets was calculated with the software and calibrated to the highly accurate average pitch of the nanomagnet array produced by the e-beam lithography tool (Vistec VB300). The thickness of the nanomagnets was determined with AFM performed in noncontact mode using the Veeco Dimension 3100 system. MFM measurements using the same instrument were conducted in a dynamic-lift mode with a lift-off distance of 30 nm.

Magneto-optic Kerr spectroscopy

To perform high-resolution magneto-optic Kerr spectroscopy, we used a focused MOKE system in lateral mode. A 635-nm diode laser was directed toward the sample, which was located between the poles of a vector magnet. The laser spot size on the sample was approximately 50 μm, covering approximately 10^4 nanomagnets. However, the magnetic field at the probe spot was calibrated by a three-axis Hall probe sensor (C-H3A-2m Three Axis Magnetic Field Transducer, SENIS GmbH). The accuracy of the magnetic field measurement is estimated at ~1%. The time to sweep full hysteresis loops was 20 min (1 Oe/s).

Energy dissipation calculation

The energy dissipation was calculated by the following equation

$$\begin{aligned}\int M \cdot \mathrm{d}H &= \int \{(M_x \times \mathrm{d}H_x) + (M_y \times \mathrm{d}H_y)\} \\ &= \int (M_x \times \mathrm{d}H_x) + \int (M_y \times \mathrm{d}H_y) \\ &= \mathrm{Area}_{m_x\text{-}H_x\mathrm{loops}} + \mathrm{Area}_{m_y\text{-}H_y\mathrm{loops}}\end{aligned}$$

The measured value of the energy dissipation is not dependent on the laser spot size/shape or the number of nanomagnets illuminated by the spot. This is because the saturation magnetization M_S and the average spin moment for the nanomagnets μ ($=M_S V$) of each individual magnet were separately measured as described below. The magnetization as measured by MOKE was calibrated by setting the saturated value of the MOKE signal to this average saturated spin moment, μ.

SUPPLEMENTARY MATERIALS

Supplementary material for this article is available at http://advances.sciencemag.org/cgi/content/full/2/3/e1501492/DC1

Fig. S1. The temperature dependence of the magnetization curves as measured by MOKE on both the easy and hard axes.

Fig. S2. *m-H* loops of the total magnetic moment of the full sample.

Fig. S3. (A) Hard axis *m-H* curves corresponding to stage 1 of the Landauer erasure protocol with various sample tilt angles.

Fig. S4. (A) The simulated energy dissipation at 0 K by varying the maximum fields ($H_{x,\ max}$ and $H_{y,\ max}$) field.

Video S1. The comprehensive hysteresis loops during the complete erasure process.

References (*14*–*16*)

REFERENCES AND NOTES

1. R. Landauer, Irreversibility and heat generation in the computing process. *IBM J. Res. Dev.* **5**, 183–191 (1961).
2. R. Landauer, Dissipation and noise immunity in computation and communication. *Nature* **335**, 779–784 (1988).
3. J. D. Meindl, J. A. Davis, The fundamental limit on binary switching energy for terascale integration (TSI). *IEEE J. Solid-St. Circ.* **35**, 1515–1516 (2000).
4. A. Bérut, A. Arakelyan, A. Petrosyan, S. Ciliberto, R. Dilenshneider, E. Lutz, Experimental verification of Landauer's principle linking information and thermodynamics. *Nature* **484**, 187–189 (2012).
5. Y. Jun, M. Gavrilov, J. Bechhoefer, High-precision test of Landauer's principle in a feedback trap. *Phys. Rev. Lett.* **113**, 190601 (2014).
6. T. Sagawa, Thermodynamics of information processing in small systems. *Prog. Theor. Phys.* **127**, 1–56 (2012).
7. C. H. Bennett, The thermodynamics of computation—A review. *Int. J. Theor. Phys.* **21**, 905–940 (1982).
8. S. Salahuddin, S. Datta, Interacting systems for self-correcting low power switching. *App. Phys. Lett.* **90**, 093503 (2007).
9. D. A. Allwood, G. Xiong, M. D. Cooke, R. P. Cowburn, Magneto-optical Kerr effect analysis of magnetic nanostructures. *J. Phys. D Appl. Phys.* **36**, 2175–2182 (2003).
10. B. Lambson, D. Carlton, J. Bokor, Exploring the thermodynamic limits of computation in integrated systems: Magnetic memory, nanomagnetic logic, and the Landauer limit. *Phys. Rev. Lett.* **107**, 010604 (2011).
11. M. Madami, M. d'Aquino, G. Gubbiotti, S. Tacchi, C. Serpico, G. Carlotti, Micromagnetic study of minimum-energy dissipation during Landauer erasure of either isolated or coupled nanomagnetic switches. *Phys. Rev. B* **90**, 104405 (2014).
12. D. C. Jiles, D. L. Atherton, Ferromagnetic hysteresis. *IEEE Trans. Magn.* **19**, 2183–2185 (1983).
13. D. A. Allwood, G. Xiong, C. C. Faulkner, D. Atkinson, D. Petit, R. P. Cowburn, Magnetic domain-wall logic. *Science* **309**, 1688–1692 (2005).
14. B. Lee, J. Hong, N. Amos, I. Dumer, D. Litvinov, S. Khizroev, Sub-10-nm-resolution electron-beam lithography toward very-high-density multilevel 3D nano-magnetic information devices. *J. Nanopart. Res.* **15**, 1665 (2013).
15. Z. Gu, M. E. Nowakowski, D. B. Carlton, R. Storz, M.-Y. Im, J. Hong, W. Chao, B. Lambson, P. Bennett, M. T. Alam, M. A. Marcus, A. Doran, A. Young, A. Scholl, J. Bokor, Sub-nanosecond signal propagation in anisotropy-engineered nanomagnetic logic chains. *Nat. Commun.* **6**, 6466 (2015).
16. R. V. Telesnin, E. N. Ilyicheva, N. G. Kanavina, N. B. Stepanova, A. G. Shishkov, Domain-wall motion in thin permalloy films in pulsed magnetic field. *IEEE Trans. Magn.* **5**, 232–236 (1969).

Acknowledgments

Funding: We acknowledge financial support from the NSF Center for Energy Efficient Electronics Science under award no. 0939514. The work at The Molecular Foundry (TMF) was supported by the U.S. Department of Energy, Office of Basic Energy Sciences, Division of Materials Sciences and Engineering under contract no. DE-AC02-05CH11231. **Author contributions:** J.B. supervised

the project. J.H. and B.L. established and performed MOKE experiments and simulation. J.H. performed AFM/MFM and SEM measurements and error analyses. S.D. fabricated nanomagnets using e-beam lithography. J.H., B.L., and J.B. wrote the manuscript with input from all authors. **Competing interests:** The authors declare that they have no competing interests. **Data and materials availability:** All data needed to evaluate the conclusions in the paper are present in the paper and/or the Supplementary Materials. Data related to this paper may be requested from the authors.

Submitted 20 October 2015
Accepted 19 January 2016
Published 11 March 2016
10.1126/sciadv.1501492

Citation: J. Hong, B. Lambson, S. Dhuey, J. Bokor, Experimental test of Landauer's principle in single-bit operations on nanomagnetic memory bits. *Sci. Adv.* **2**, e1501492 (2016).

ScienceAdvances
AAAS

Experimental test of Landauer's principle in single-bit operations on nanomagnetic memory bits

Jeongmin Hong, Brian Lambson, Scott Dhuey and Jeffrey Bokor

(March 11, 2016)
Sci Adv 2016, 2:.
doi: 10.1126/sciadv.1501492

VOLUME 34 · NUMBER 35 · DECEMBER 10, 2016

JOURNAL OF CLINICAL ONCOLOGY

REVIEW ARTICLE

Obesity and Other Cancers

Lin Yang, Bettina F. Drake, and Graham A. Colditz

Lin Yang, Bettina F. Drake, and Graham A. Colditz, Washington University School of Medicine and Siteman Cancer Center, St Louis, MO; and Lin Yang, Center for Public Health, Medical University of Vienna, Vienna, Austria.

Published online ahead of print at www.jco.org on November 7, 2016.

Authors' disclosures of potential conflicts of interest are found in the article online at www.jco.org. Author contributions are found at the end of this article.

Corresponding author: Graham A. Colditz, MD, DrPH, Washington University School of Medicine, 660 S Euclid Ave, Campus Box 8100, St Louis, MO 63110; e-mail: colditzg@wustl.edu.

0732-183X/16/3435w-4231w/$20.00

DOI: 10.1200/JCO.2016.68.4837

A B S T R A C T

Purpose

Evidence on overweight, obesity, and an increased risk of cancer continues to accumulate and was updated in the 2016 handbook on weight control from the International Agency for Research on Cancer (IARC). The underlying primary data, together with dose-response meta-analysis and, finally, pooled analysis of individual participant data, add insight into the relation between obesity and cancer risk and prognosis. We summarize the evidence for mortality from prostate cancer, hematologic malignancies, and kidney cancer.

Methods

We reviewed pooled analysis of rare end points across cohorts, regardless of primary results reported from the individual studies, further reducing risk of publication bias. Of these cancer sites, only kidney cancer was included in the IARC 2002 report, although mortality from prostate cancer and hematologic malignancies was noted in the American Cancer Society prospective cohort study in 2003. The 2016 update from the IARC added details for prostate and hematologic malignancies, classifying the evidence as sufficient to conclude that avoiding excess body fatness lowers the risk of multiple myeloma but found that the evidence for it lowering the risk of prostate cancer mortality or diffuse large B-cell lymphoma was limited.

Results

A higher body mass index is associated with an increased risk of advanced prostate cancer and prostate cancer mortality and is associated with worse survival in most subtypes of hematologic malignancies, in a dose-response fashion. Evidence for kidney cancer is built mostly on retrospective data, which supports an obesity paradox in patients with the clear cell variant; however, population-based cohort data indicate that a higher cohort-entry body mass index is associated with worse kidney cancer–specific survival.

Conclusion

Together, these data add support to the evidence for a growing cancer burden caused by adiposity in both early adult and later adult life, yet leave open the question of the means of weight management after diagnosis as a strategy to improve survival.

J Clin Oncol 34:4231-4237. © 2016 by American Society of Clinical Oncology

INTRODUCTION

Here we review the current research addressing the role of adiposity/obesity in mortality and prognostic outcomes in prostate, hematologic, and renal cancers. Excess weight or adiposity is a common risk factor for cancer, progression, and nonsurvival; this provides a unique opportunity to address a modifiable risk factor through primary and secondary interventions. Evidence reviewed by the International Agency for Research on Cancer (IARC) and the World Cancer Research Fund support the finding that being overweight or obese is an established cause of several cancers including breast, endometrium, esophagus (adenocarcinoma), renal, and colon and rectum.[1] Since the IARC report in 2002, evidence has expanded to suggest an association between a higher body mass index (BMI) and advanced prostate or prostate cancer mortality and several hematologic cancers.[2] The 2016 IARC update on body fatness and cancer concluded that the evidence was sufficient to conclude that avoiding excess body fatness lowers the risk of multiple myeloma but evidence for its lowering the risk of prostate cancer mortality and diffuse large B-cell lymphoma is limited. Thus, for these cancer sites, the committee could not rule out bias and confounding as contributing to the positive associations observed.[2] In addition, excessive weight is a risk factor for cancer mortality overall.[3] To

first summarize the current evidence on cause, we placed the greatest emphasis on prospective cohort studies reported as pooled analyses of individual participant data. This approach reduces variation in analytic approaches among studies. Furthermore, for rare cancers such a multiple myeloma, it allows all cohorts to contribute end points regardless of whether they have published results or not, hence reducing the publication bias that can distort meta-analyses limited to results already published in the literature. In this review, we summarize studies on obesity and the incidence of prostate, hematologic, and renal cancers. We are not aware of any randomized controlled trials of weight-loss interventions after diagnosis of cancer at these sites. We summarize the observational evidence on adiposity/obesity and cancer prognosis in Table 1, and describe associations for each included cancer site in greater detail.

Table 1. Summary of Observational Studies on Adiposity/Obesity and Cancer Prognosis

Cancer type	Prospective Cohort*	Patient Cohort† and Retrospective Cohort‡
Prostate cancer		
Cancer-specific survival	Meta-analysis[4]: RR, 1.15 (95% CI, 1.06 to 1.25) per 5 kg/m^2 increased BMI	Meta-analysis[4]: RR, 1.20 (95% CI, 0.99 to 1.46) per 5 kg/m^2 increased BMI
Hematologic malignancies		
Lymphoma		
Cancer-specific survival	Meta-analysis[5]: RR, 1.14 (95% CI, 1.04 to 1.26) per 5 kg/m^2 increased BMI	
Leukemia		
Overall survival	Meta-analysis[6]: RR, 1.29 (95% CI, 1.11 to 1.49) in obese (BMI $\geq$ 30 kg/m^2) *v* normal- weight (18.5 kg/m^2 $\leq$ BMI $<$ 25 kg/m^2) patients RR_{men}, 1.45 (95% CI, 1.22 to 1.72) RR_{women}, 1.14 (95% CI, 0.99 to 1.33)	Pooled patient cohort[7]: HR, 1.72 (95% CI, 1.15 to 2.58) in obese (BMI $\geq$ 30 kg/m^2) *v* nonobese (BMI $<$ 30 kg/m^2) patients with APL
Relapse		Cohort of patients with APL[8]: HR, 2.45 (95% CI, 1.00 to 5.99) in overweight/obese (BMI $\geq$ 25 kg/m^2) *v* under-/normal-weight (BMI $<$ 25 kg/m^2) patients
Pediatric leukemia		
Overall survival		Meta-analysis[9]: HR, 1.30 (95% CI, 1.16 to 1.46) in obese (BMI $>$ 95%) *v* nonobese (BMI $\leq$ 95%) patients
Event-free survival		Meta-analysis[9]: HR, 1.46 (95% CI, 1.29 to 1.64) obese (BMI $>$ 95%) *v* nonobese (BMI $\leq$ 95%) patients
Multiple myeloma		
Cancer-specific survival	Meta-analysis[10]: RR, 1.15 (95% CI, 1.04 to 1.27) in overweight (25.0 kg/m^2 $\leq$ BMI $<$ 30 kg/m^2) patients; RR, 1.54 (95% CI, 1.35 to 1.76) in obese (BMI $\geq$ 30 kg/m^2) *v* normal-weight (18.5 kg/m^2 $\leq$ BMI $<$ 25 kg/m^2) patients Pooled analysis[11]: HR, 1.09 (95% CI, 1.03 to 1.16) per 5 kg/m^2 increased cohort-entry BMI; HR, 1.22 (95% CI, 1.09 to 1.35) per 5 kg/m^2 increased early-adulthood BMI; HR, 1.06 (95% CI, 1.02 to 1.10) per 5 cm increased waist circumference.	
Kidney cancer		
Overall survival		Patient cohort[12]: HR, 0.50 (95% CI, 0.31 to 0.81) in obese (30 kg/m^2 $\leq$ BMI $<$ 35 kg/m^2) patients; HR, 0.24 (95% CI, 0.09 to 0.60) in severely obese (BMI $\geq$ 35 kg/m^2) *v* nonobese (BMI kg/m^2 $<$ 30 kg/m^2) patients Meta-analysis of retrospective studies[13]: HR, 0.57 (95% CI, 0.43 to 0.76) in highest *v* lowest BMI category
Cancer-specific survival	Million Women cohort[14]: RR, 1.65 (95% CI, 1.28 to 2.13) per 10 kg/m^2 increased BMI	Patient cohort[15]: HR, 0.87 (95% CI, 0.82 to 0.94) per 1 kg/m^2 increased BMI in patients with clear cell variant; HR, 1.32 (95% CI, 1.03 to 1.70) per 1 kg/m^2 increased BMI in patients with chromophobe variant; no association in patients with papillary variant Meta-analysis of retrospective studies[13]: HR, 0.59 (95% CI, 0.48 to 0.74) in highest *v* lowest BMI category
Recurrence-free survival		Meta-analysis of retrospective studies[13]: HR, 0.49 (95% CI, 0.30 to 0.81) in highest *v* lowest BMI category

Abbreviations: APL, acute promyelocytic leukemia; BMI, body mass index; HR, hazard ratio; RR, relative risk.
*Cancer-free at cohort entry, BMI measured before diagnosis.
†BMI measured at diagnosis.
‡BMI measured after diagnosis.

OBESITY AND INCIDENCE OF PROSTATE, HEMATOLOGIC, AND RENAL CANCERS

There is no clear link between obesity and overall prostate cancer incidence. A meta-analysis of 27 cohorts estimated the association between obesity and prostate cancer risk, reporting a relative risk of 1.03 [95% CI, 1.00 to 1.07] per 5 kg/m^2 increase in BMI ($P = .11$) with varied results across cohort studies.[16] A growing body of research indicates that obesity is associated with an increased risk of advanced prostate cancer.[17,18] The IARC update concluded that the evidence for an association between excess body weight and fatal prostate cancer was limited.[2] Meta-analytic data of prospective cohort studies suggest a modest but consistent direct effect of BMI on the incidence of lymphoma, multiple myeloma, and adult leukemia.[16,5,10,19] Risk increases in a dose-response fashion. It is also important to note that the adverse impact of BMI on the risk of lymphoma and multiple myeloma may start growing early, during young adulthood. On the basis of a review of the evidence, the IARC stated in 2016 that the evidence was sufficient to conclude that an absence of excess body fatness lowers the risk of multiple myeloma. Mounting evidence for kidney cancer continues to show increasing risk with increasing BMI in a dose-response fashion.[16,20,21]

OBESITY AND PROSTATE CANCER PROGNOSIS

A growing body of research indicates that obesity is associated with worse pathologic outcomes in prostate cancer; however, evidence of an association has varied across cohort studies. Analysis of 57 European and American prospective studies shows an overall RR of prostate cancer mortality of 1.13 (95% CI, 1.02 to 1.24) for a 5 kg/m^2 increase in BMI.[22] In contrast, in an analysis of Asian cohorts, obesity was not related to prostate cancer mortality.[23] Obesity is linked to hormonal imbalances and hormonal factors that may influence prostate cancer progression.[24,25] Several mechanisms that may contribute to the growth of prostate cancer, including adiponectin levels[26] and inflammatory mediators,[27] have also been explored for relations with survival and mortality.

Epidemiologic studies show that obesity is associated with an increased risk of prostate cancer aggressiveness,[28-32] progression,[32-36] and cancer-specific mortality.[27,32,37-39] Chalfin et al[40] found that obese men are more likely to have worse pathology and a higher risk of recurrence of their cancer. Furthermore, Asmar et al[41] and Ly et al[42] reported that increased BMI was significantly associated with a higher risk of biochemical failure.

Incidence rates for prostate cancer are 60% higher for African-American men,[43] who also have the highest prostate cancer mortality. Prostate cancer accounts for 44% of the overall cancer mortality disparity between African-American and white men.[44] Studies suggest that obesity may contribute to this disparity in prostate cancer outcomes.[31,4,45-47] African-American men have higher rates of obesity compared with white men.[48] Obesity is also associated with both prostate tumor biology and a delayed prostate-specific antigen diagnosis of prostate cancer. Studies have also found that in men with low-risk disease, obese African-American men have a higher risk of recurrence compared with obese white men.[49]

Although physical activity, especially vigorous activity, is inversely related to the incidence of aggressive prostate cancer[50-56] and may also be associated with a decreased overall incidence of prostate cancer,[57] associations between obesity and prostate cancer seem to be independent of the level of physical activity.

Continued study is warranted to clarify the associations reported to date and to identify additional factors that may modify the relationship between obesity and prostate cancer prognosis. Short follow-up and lack of control for smoking are major limitations in many of these studies, especially in data from clinical studies. Other neglected areas of study include the timing of BMI measurements (eg, 1, 2, or 5 years before prostate cancer diagnosis) and the ability to assess ethnic disparities, including the association between obesity and lethal prostate cancer in African-American and other minority groups.

OBESITY AND HEMATOLOGIC MALIGNANCIES PROGNOSIS

Epidemiologic evidence from prospective cohort studies indicates does-response associations between excessive body weight and an increased risk of mortality from several subtypes of hematologic malignancies.[16]

The most recent meta-analysis of BMI and lymphoma by Larsson and Wolk[5] summarized available prospective data through May 2011, including five studies (n = 3,407 cases) of BMI in relation to non-Hodgkin lymphoma (NHL) mortality. A does-response association between increasing BMI and increased NHL mortality was observed. Per 5 kg/m^2 increase in BMI, the RR of NHL mortality was 1.14 (95% CI, 1.04 to 1.26). Conversely, some studies have reported that a higher BMI is associated with better survival outcomes in retrospective data.[58,59] In a more recent study, Hwang et al[60] reported data for Asian patients with diffuse large B-cell lymphoma and noted worse overall survival (hazard ratio [HR], 1.29; 95% CI, 1.08 to 7.95) and worse progression-free survival (HR, 2.59; 95% CI, 1.06 to 6.35) in the obese. Similarly, Leo et al[61] observed an association between obesity and worse NHL-specific survival in patients of diverse ethnicities (HR, 1.77; 95% CI, 1.30 to 2.41). One prospective study in Taiwan included a measure of central obesity (defined as waist circumference $\geq$ 90 cm in men, $\geq$ 80 cm in women) and reported worse NHL-specific survival in centrally obese patients (HR, 2.16; 95% CI, 1.41 to 3.31) after adjusting for BMI.[62] Across these studies, a worse prognostic outcome in underweight patients was reported, consistent with the impact of the disease process on weight.[60,63,64]

Multiple myeloma is relatively rare compared with lymphoma or leukemia; it has historically poor survival, with a 5-year survival rate of $<$ 50% in the United States.[65] Both overweight and obesity are associated with multiple myeloma mortality. Wallin and Larsson[10] combined available data restricted to prospective cohort studies published up to 2011. Data from five prospective cohorts (n = 1,845 cases) suggested worse overall survival in both overweight (RR, 1.15; 95% CI, 1.04 to 1.27) and obese (RR, 1.54; 95% CI, 1.35 to 1.76) patients with multiple myeloma. These associations, assessed as a dose-response per 5 kg/m^2 increase in BMI, gave a 21% increased risk of multiple myeloma mortality per 5 kg/m^2 increase in BMI. The most updated pooled analysis of individual participant data from 20 prospective studies[11] extended the

analysis on multiple myeloma mortality, including the association of early-adulthood BMI and weight distribution. BMI at cohort entry (RR, 1.09 [95% CI, 1.03 to 1.16] per 5 kg/m^2) and higher early-adulthood BMI (RR, 1.22 [95% CI, 1.09 to 1.35] per 5 kg/m^2) were both associated with increased multiple myeloma mortality. In this combined analysis, waist circumference (HR, 1.06 [95% CI, 1.02 to 1.10] per 5 cm) was also associated with mortality, suggesting the deleterious impact of central obesity. The association between early-adulthood BMI and mortality was stronger in women (HR, 1.95 [95% CI, 1.33 to 2.86] for BMI $>$ 25 kg/m^{2}compared with BMI of 18.5 to 25 kg/m^2).

Leukemia accounts for 2.5% of all cancer cases globally.[66] Among its four major types, acute lymphoblastic leukemia (ALL) is the most common tumor in children, whereas other subtypes (chronic lymphocytic leukemia, acute myeloid leukemia, and chronic myeloid leukemia) occur mostly later in adult life. In adult populations, obesity is associated with a higher risk of leukemia mortality (RR, 1.29; 95% CI, 1.11 to 1.49) on the basis of data from six prospective cohort studies (n = 2,358 deaths) up to 2011.[6] The association between elevated BMI and leukemia mortality was linear in both men (1.9% per kg/m^2, P = .10) and women (1.2% per kg/m^2, P = .01). Several studies were conducted with retrospective design after the meta-analysis. Wenzell et al[67] reported improved overall survival in obese patients with leukemia, but others reported null findings.[68,69] A recent analysis that pooled data from four Cancer and Leukemia Group B (Alliance) clinical trials of patients with acute myeloid leukemia, including acute promyelocytic leukemia (APL), linked obesity to worse overall survival (HR, 1.72; 95% CI, 1.15 to 2.58) and disease-free survival (HR, 1.53; 95% CI, 1.03 to 2.27) in APL, but not in non-APL, suggesting a possible distinct biologic relation between obesity in APL.[7]

Studies of childhood leukemia have been focused mostly on ALL, the most common tumor in children. Data on other subtypes of childhood leukemia are scarce. Amankwah et al[9] conducted a meta-analysis synthesizing data from 11 studies up to 2015 to access the association between obesity at diagnosis and pediatric ($<$ 21 years old) acute leukemia survival and relapse. They reported worse overall survival (HR, 1.30; 95% CI, 1.16 to 1.46) and worse event-free survival (HR, 1.46; 95% CI, 1.29 to 1.64) in patients who were obese at diagnosis. When the analysis was restricted to ALL only, both these associations persisted and were stronger (HR, 2.25 [95% CI, 1.33 to 3.82] for overall survival; and HR, 1.49 [95% CI, 1.30 to 1.71] for event-free survival).

Current evidence suggests a modest but consistent direct effect of BMI on poor prognostic outcomes in lymphoma, multiple myeloma, and leukemia. Risk increases in a dose-response manner in several subtypes of hematologic malignancies. It is also important to note that the adverse impact of BMI on multiple myeloma prognosis may reflect etiologic risk that starts growing early, during young adulthood. Separating the etiologic effect, disease severity at diagnosis, and outcomes requires more evidence from randomized interventions. To date, the small number of prospective studies and the lack of adjustment for possible confounding factors such as treatment, smoking, and comorbidities may bias the estimates reported for mortality as a result of hematologic malignancies.

OBESITY AND RENAL CANCER PROGNOSIS

Renal cancer (predominantly renal cell carcinoma [RCC]) accounts for approximately 2% of new cancer diagnoses. The IARC review in 2002 included evidence from four cohort studies and concluded a consistently positive association,[1] as did the World Cancer Research Fund review in 2007[70] and the subsequent update in 2015. The IARC 2016 update continues to classify RCC as caused by obesity.[2] Obesity is, however, associated with improved prognosis in patients with RCC, supported by a 2013 systematic review. This meta-analysis included data from 20 studies and reported improved overall (HR, 0.57; 95% CI, 0.43 to 0.76), cancer-specific (HR, 0.59; 95% CI, 0.48 to 0.74), and recurrence-free (HR, 0.49; 95% CI, 0.30 to 0.81) survival in patients with higher BMI than in those with lower BMI.[13] Notably, all these studies used a retrospective design. Data from prospective cohorts of patients are generally sparse. A 2016 study analyzed data from a prospective randomized trial of 845 localized high-risk patients with RCC and reported that a higher BMI was associated with improved survival.[12] The UK Million Women Study cohort found worse kidney-specific survival (RR, 1.65; 95% CI, 1.28 to 2.13) for every 10 kg/m^2 increment in cohort entry BMI, and the risk with increasing BMI was stronger in never-smokers.[71] Recently, a Korean patient cohort of 2,769 cases investigated the association between BMI and prognosis of nonmetastatic RCC by histologic subtype.[15] This study reported that a higher BMI ($\geq$ 23 kg/m^2) was associated with improved cancer-specific survival (HR, 0.87) in patients with the clear cell variant, but worse cancer-specific (HR, 1.32) and recurrence-free (HR, 1.32) survival in patients with the chromophobe variant. No association was seen in patients with the papillary variant.[15] The association between obesity and RCC prognosis thus may differ by histologic subtype, although further study is needed to confirm these associations.

Current evidence on RCC prognosis is based mostly on data generated by retrospective studies, with few data available from prospective cohorts. These data support an obesity paradox in patients with RCC with the clear cell variant. That is, although obesity is an established risk factor for RCC, being obese at diagnosis seems to be associated with a favorable outcome compared with the outcomes of patients with normal weight at diagnosis. Some studies have reported that obese patients are more likely to be diagnosed with favorable clinical characteristics, including lower-stage disease, lower Fuhrman grade, and smaller tumors, compared with normal-weight patients.[13] Further support from gene expression analyses demonstrates that obese patients are less likely to have a tumor with fatty acid synthase (FASN), which encodes the proteins necessary for tumor growth.[72]

OBESITY AND CANCER PROGNOSIS MECHANISMS

The mechanisms of obesity in relation to cancer development and prognosis are poorly understood on the basis of current studies. Furthermore, determining how much of an outcome is attributable to obesity, treatment, or the underlying disease subtypes/biology is difficult. It has been proposed that chemotherapy dosing that is based on body surface area in obese patients may be biased and may

result in poor survival. Hourdequin et al[73] conducted a meta-analysis and reported similar or lower levels of toxicity in obese patients compared with normal-weight patients when they received body surface area–based chemotherapy dosing, and no differences in survival outcomes were seen. Obesity reflects increased adipose tissue, which secretes a range of adipokines that may play a role in cancer development. These adipokines include inflammatory cytokines such as tumor necrosis factor-α and interleukin-6, which likely cause a chronic inflammatory microenvironment that also promotes cancer progrssion.[74] Such inflammation may lead to the activation of transcription factor nuclear factor-kB, which is linked to several cancers, including lymphoma.[75] Adipose tissue secretes leptin and adiponectin, which are both linked to prostate cancer and RCC.[75] Other pathways associated with obesity, such as hyperinsulinemia, insulin-insulin-like growth factor signaling, and lipid levels, have been explored in relation to cancer progression. For example, downgraded expression of FASN was found in obese patients with RCC and was associated with an improved survival.[72] In patients with prostate cancer, overweight men carrying a variant FASN allele had a poorer prognosis compared with men of normal weight.[76] These possible pathways require further research to address the potential response to weight loss after cancer diagnosis and possible improvement in survival outcomes. The cancer sites in this review have less evidence for a benefit of weight loss after diagnosis than is currently accumulating for breast and colorectal cancer.

CLINICAL RECOMMENDATIONS

At the current time, the rapidly rising prevalence of obesity in all age groups may contribute to a large future cancer burden. The American Society of Clinical Oncology established a multipronged initiative and committed to reducing the impact of obesity on cancer.[77] The delivery of effective and efficient counseling on weight-management strategies, however, might be challenged by the poor understanding of the relationship between obesity and cancer progression. Cancer survivors are encouraged to achieve and maintain a healthy weight. Guidelines from the American Cancer Society[78] and the American College of Sports Medicine[79] suggest that cancer survivors should follow the physical activity guidelines for Americans, with specific exercise programming adaptations on the basis of disease- and treatment-related adverse effects (ie, ≥ 150 minutes per week of moderate-intensity physical activity). In addition, cancer survivors are encouraged to achieve a high intake of vegetables, fruits, and whole grains and to avoid high-calorie foods and beverages. It is still unclear what impact specific lifestyle components (energy intake/diet or energy expenditure/physical activity) that contribute to obesity have in relation to cancer progression. With limited prospective longitudinal data on weight change in relation to prognostic outcomes, and in the absence of randomized controlled trials, cancer survivors should probably not be advised to voluntarily lose weight to reduce cancer recurrence or mortality. Yet maintaining a healthy lifestyle through physical activity, healthy diet, and avoiding weight gain may lead to general health benefits beyond cancer-specific outcomes.

FUTURE DIRECTIONS

Future research to advance the understanding of weight management in the cancer care continuum can help document the time course of weight loss and improved outcomes after diagnosis and identify pathways that may be addressed more directly through drug therapies. Given the association between obesity and cancer incidence and mortality, shifting the perspective of clinical oncology practice to include weight management as a component of primary and secondary prevention of cancer is increasingly important.

AUTHORS' DISCLOSURES OF POTENTIAL CONFLICTS OF INTEREST

Disclosures provided by the authors are available with this article at www.jco.org.

AUTHOR CONTRIBUTIONS

Conception and design: Lin Yang, Graham A. Colditz
Administrative support: Graham A. Colditz
Collection and assembly of data: All authors
Data analysis and interpretation: Lin Yang, Graham A. Colditz
Manuscript writing: All authors
Final approval of manuscript: All authors
Accountable for all aspects of the work: All authors

REFERENCES

1. International Agency for Research on Cancer: Weight Control and Physical Activity. Lyon, France, International Agency for Research on Cancer, 2002
2. Lauby-Secretan B, Scoccianti C, Loomis D, et al: Body fatness and cancer–viewpoint of the IARC Working Group. N Engl J Med 375:794-798, 2016
3. Calle EE, Rodriguez C, Walker-Thurmond K, et al: Overweight, obesity, and mortality from cancer in a prospectively studied cohort of U.S. adults. N Engl J Med 348:1625-1638, 2003
4. Cao Y, Ma J: Body mass index, prostate cancer-specific mortality, and biochemical recurrence: A systematic review and meta-analysis. Cancer Prev Res (Phila) 4:486-501, 2011
5. Larsson SC, Wolk A: Body mass index and risk of non-Hodgkin's and Hodgkin's lymphoma: A meta-analysis of prospective studies. Eur J Cancer 47: 2422-2430, 2011
6. Castillo JJ, Reagan JL, Ingham RR, et al: Obesity but not overweight increases the incidence and mortality of leukemia in adults: A meta-analysis of prospective cohort studies. Leuk Res 36:868-875, 2012
7. Castillo JJ, Mulkey F, Geyer S, et al: Relationship between obesity and clinical outcome in adults with acute myeloid leukemia: A pooled analysis from four CALGB (alliance) clinical trials. Am J Hematol 91:199-204, 2016
8. Breccia M, Mazzarella L, Bagnardi V, et al: Increased BMI correlates with higher risk of disease relapse and differentiation syndrome in patients with acute promyelocytic leukemia treated with the AIDA protocols. Blood 119:49-54, 2012
9. Amankwah EK, Saenz AM, Hale GA, et al: Association between body mass index at diagnosis and pediatric leukemia mortality and relapse: A systematic review and meta-analysis. Leuk Lymphoma 57:1140-1148, 2016
10. Wallin A, Larsson SC: Body mass index and risk of multiple myeloma: A meta-analysis of prospective studies. Eur J Cancer 47:1606-1615, 2011
11. Teras LR, Kitahara CM, Birmann BM, et al: Body size and multiple myeloma mortality: A pooled

analysis of 20 prospective studies. Br J Haematol 166:667-676, 2014

12. Donin NM, Pantuck A, Klöpfer P, et al: Body mass index and survival in a prospective randomized trial of localized high-risk renal cell carcinoma. Cancer Epidemiol Biomarkers Prev 25:1326-1332, 2016

13. Choi Y, Park B, Jeong BC, et al: Body mass index and survival in patients with renal cell carcinoma: A clinical-based cohort and meta-analysis. Int J Cancer 132:625-634, 2013

14. Reeves GK, Pirie K, Beral V, et al: Cancer incidence and mortality in relation to body mass index in the Million Women Study: Cohort study. BMJ 335: 1134, 2007

15. Lee WK, Hong SK, Lee S, et al: Prognostic value of body mass index according to histologic subtype in nonmetastatic renal cell carcinoma: A large cohort analysis. Clin Genitourin Cancer 13: 461-468, 2015

16. Renehan AG, Tyson M, Egger M, et al: Body-mass index and incidence of cancer: A systematic review and meta-analysis of prospective observational studies. Lancet 371:569-578, 2008

17. De Nunzio C, Albisinni S, Freedland SJ, et al: Abdominal obesity as risk factor for prostate cancer diagnosis and high grade disease: A prospective multicenter Italian cohort study. Urol Oncol 31: 997-1002, 2013

18. Tewari R, Rajender S, Natu SM, et al: Significance of obesity markers and adipocytokines in high grade and high stage prostate cancer in North Indian men - a cross-sectional study. Cytokine 63:130-134, 2013

19. Larsson SC, Wolk A: Overweight and obesity and incidence of leukemia: A meta-analysis of cohort studies. Int J Cancer 122:1418-1421, 2008

20. Adams KF, Leitzmann MF, Albanes D, et al: Body size and renal cell cancer incidence in a large US cohort study. Am J Epidemiol 168:268-277, 2008

21. Pischon T, Lahmann PH, Boeing H, et al: Body size and risk of renal cell carcinoma in the European Prospective Investigation into Cancer and Nutrition (EPIC). Int J Cancer 118:728-738, 2006

22. Whitlock G, Lewington S, Sherliker P, et al: Body-mass index and cause-specific mortality in 900 000 adults: Collaborative analyses of 57 prospective studies. Lancet 373:1083-1096, 2009

23. Fowke JH, McLerran DF, Gupta PC, et al: Associations of body mass index, smoking, and alcohol consumption with prostate cancer mortality in the Asia Cohort Consortium. Am J Epidemiol 182: 381-389, 2015

24. Burton AJ, Tilling KM, Holly JM, et al: Metabolic imbalance and prostate cancer progression. Int J Mol Epidemiol Genet 1:248-271, 2010

25. Presti JC, Jr: Obesity and prostate cancer. Curr Opin Urol 15:13-16, 2005

26. Li H, Stampfer MJ, Mucci L, et al: A 25-year prospective study of plasma adiponectin and leptin concentrations and prostate cancer risk and survival. Clin Chem 56:34-43, 2010

27. Ma J, Li H, Giovannucci E, et al: Prediagnostic body-mass index, plasma C-peptide concentration, and prostate cancer-specific mortality in men with prostate cancer: A long-term survival analysis. Lancet Oncol 9:1039-1047, 2008

28. MacInnis RJ, English DR: Body size and composition and prostate cancer risk: Systematic review and meta-regression analysis. Cancer Causes Control 17:989-1003, 2006

29. Bañez LL, Hamilton RJ, Partin AW, et al: Obesity-related plasma hemodilution and PSA concentration among men with prostate cancer. JAMA 298:2275-2280, 2007

30. Loeb S, Yu X, Nadler RB, et al: Does body mass index affect preoperative prostate specific antigen velocity or pathological outcomes after radical prostatectomy? J Urol 177:102-106, discussion 106, 2007

31. Su LJ, Arab L, Steck SE, et al: Obesity and prostate cancer aggressiveness among African and Caucasian Americans in a population-based study. Cancer Epidemiol Biomarkers Prev 20:844-853, 2011

32. Allott EH, Masko EM, Freedland SJ: Obesity and prostate cancer: Weighing the evidence. Eur Urol 63:800-809, 2013

33. Strom SS, Kamat AM, Gruschkus SK, et al: Influence of obesity on biochemical and clinical failure after external-beam radiotherapy for localized prostate cancer. Cancer 107:631-639, 2006

34. Davies BJ, Smaldone MC, Sadetsky N, et al: The impact of obesity on overall and cancer specific survival in men with prostate cancer. J Urol 182:112-117, 2009; discussion 117

35. Bassett WW, Cooperberg MR, Sadetsky N, et al: Impact of obesity on prostate cancer recurrence after radical prostatectomy: Data from CaPSURE. Urology 66:1060-1065, 2005

36. Freedland SJ, Aronson WJ, Kane CJ, et al: Impact of obesity on biochemical control after radical prostatectomy for clinically localized prostate cancer: A report by the Shared Equal Access Regional Cancer Hospital database study group. J Clin Oncol 22: 446-453, 2004

37. Giovannucci E, Liu Y, Platz EA, et al: Risk factors for prostate cancer incidence and progression in the health professionals follow-up study. Int J Cancer 121:1571-1578, 2007

38. Rodriguez C, Freedland SJ, Deka A, et al: Body mass index, weight change, and risk of prostate cancer in the Cancer Prevention Study II Nutrition Cohort. Cancer Epidemiol Biomarkers Prev 16:63-69, 2007

39. Wright ME, Chang SC, Schatzkin A, et al: Prospective study of adiposity and weight change in relation to prostate cancer incidence and mortality. Cancer 109:675-684, 2007

40. Chalfin HJ, Lee SB, Jeong BC, et al: Obesity and long-term survival after radical prostatectomy. J Urol 192:1100-1104, 2014

41. Asmar R, Beebe-Dimmer JL, Korgavkar K, et al: Hypertension, obesity and prostate cancer biochemical recurrence after radical prostatectomy. Prostate Cancer Prostatic Dis 16:62-66, 2013

42. Ly D, Reddy CA, Klein EA, et al: Association of body mass index with prostate cancer biochemical failure. J Urol 183:2193-2199, 2010

43. American Cancer Society. Cancer Facts & Figures 2008. Atlanta, GA, American Cancer Society, 2008

44. American Cancer Society. Cancer Facts & Figures for African Americans 2011-2012. Atlanta, GA, American Cancer Society, 2011

45. Motamedinia P, Korets R, Spencer BA, et al: Body mass index trends and role of obesity in predicting outcome after radical prostatectomy. Urology 72:1106-1110, 2008

46. Keto CJ, Aronson WJ, Terris MK, et al: Obesity is associated with castration-resistant disease and metastasis in men treated with androgen deprivation therapy after radical prostatectomy: Results from the SEARCH database. BJU Int 110:492-498, 2012

47. Joshu CE, Mondul AM, Menke A, et al: Weight gain is associated with an increased risk of prostate cancer recurrence after prostatectomy in the PSA era. Cancer Prev Res (Phila) 4:544-551, 2011

48. Flegal KM, Carroll MD, Ogden CL, et al: Prevalence and trends in obesity among US adults, 1999-2008. JAMA 303:235-241, 2010

49. Caire AA, Sun L, Polascik TJ, et al: Obese African-Americans with prostate cancer (T1c and a prostate-specific antigen, PSA, level of <10 ng/mL) have higher-risk pathological features and a greater risk of PSA recurrence than non-African-Americans. BJU Int 106:1157-1160, 2010

50. Johnsen NF, Tjønneland A, Thomsen BL, et al: Physical activity and risk of prostate cancer in the European Prospective Investigation into Cancer and Nutrition (EPIC) cohort. Int J Cancer 125:902-908, 2009

51. Giovannucci E, Leitzmann M, Spiegelman D, et al: A prospective study of physical activity and prostate cancer in male health professionals. Cancer Res 58:5117-5122, 1998

52. Giovannucci EL, Liu Y, Leitzmann MF, et al: A prospective study of physical activity and incident and fatal prostate cancer. Arch Intern Med 165: 1005-1010, 2005

53. Nilsen TI, Romundstad PR, Vatten LJ: Recreational physical activity and risk of prostate cancer: A prospective population-based study in Norway (the HUNT study). Int J Cancer 119:2943-2947, 2006

54. Patel AV, Rodriguez C, Jacobs EJ, et al: Recreational physical activity and risk of prostate cancer in a large cohort of U.S. men. Cancer Epidemiol Biomarkers Prev 14:275-279, 2005

55. Orsini N, Bellocco R, Bottai M, et al: A prospective study of lifetime physical activity and prostate cancer incidence and mortality. Br J Cancer 101:1932-1938, 2009

56. Wolin KY, Stoll C: Physical activity and urologic cancers. Urol Oncol 30:729-734, 2012

57. Liu Y, Hu F, Li D, et al: Does physical activity reduce the risk of prostate cancer? A systematic review and meta-analysis. Eur Urol 60:1029-1044, 2011

58. Carson KR, Bartlett NL, McDonald JR, et al: Increased body mass index is associated with improved survival in United States veterans with diffuse large B-cell lymphoma. J Clin Oncol 30:3217-3222, 2012

59. Weiss L, Melchardt T, Habringer S, et al: Increased body mass index is associated with improved overall survival in diffuse large B-cell lymphoma. Ann Oncol 25:171-176, 2014

60. Hwang HS, Yoon DH, Suh C, et al: Body mass index as a prognostic factor in Asian patients treated with chemoimmunotherapy for diffuse large B cell lymphoma, not otherwise specified. Ann Hematol 94:1655-1665, 2015

61. Leo QJ, Ollberding NJ, Wilkens LR, et al: Obesity and non-Hodgkin lymphoma survival in an ethnically diverse population: The Multiethnic Cohort study. Cancer Causes Control 25:1449-1459, 2014

62. Chu DM, Wahlqvist ML, Lee MS, et al: Central obesity predicts non-Hodgkin's lymphoma mortality and overall obesity predicts leukemia mortality in adult Taiwanese. J Am Coll Nutr 30:310-319, 2011

63. Park S, Han B, Cho JW, et al: Effect of nutritional status on survival outcome of diffuse large B-cell lymphoma patients treated with rituximab-CHOP. Nutr Cancer 66:225-233, 2014

64. Navarro WH, Loberiza FR, Jr., Bajorunaite R, et al: Effect of body mass index on mortality of patients with lymphoma undergoing autologous hematopoietic cell transplantation. Biol Blood Marrow Transplant 12:541-551, 2006

65. Siegel RL, Miller KD, Jemal A: Cancer statistics, 2016. CA Cancer J Clin 66:7-30, 2016

66. Rodriguez-Abreu D, Bordoni A, Zucca E: Epidemiology of hematological malignancies. Ann Oncol 18:i3-i8, 2007

67. Wenzell CM, Gallagher EM, Earl M, et al: Outcomes in obese and overweight acute myeloid leukemia patients receiving chemotherapy dosed according to actual body weight. Am J Hematol 88: 906-909, 2013

68. Tavitian S, Denis A, Vergez F, et al: Impact of obesity in favorable-risk AML patients receiving intensive chemotherapy. Am J Hematol 91:193-198, 2016

69. Medeiros BC, Othus M, Estey EH, et al: Impact of body-mass index on the outcome of adult patients with acute myeloid leukemia. Haematologica 97:1401-1404, 2012

70. World Cancer Research Fund. Food, Nutrition, Physical Activity, and the Prevention of Cancer: A Global Perspective. Washington, DC, AICR, 2007

71. Beral V, Bull D, Green J, et al: Ovarian cancer and hormone replacement therapy in the Million Women Study. Lancet 369:1703-1710, 2007

72. Hakimi AA, Furberg H, Zabor EC, et al: An epidemiologic and genomic investigation into the obesity paradox in renal cell carcinoma. J Natl Cancer Inst 105:1862-1870, 2013

73. Hourdequin KC, Schpero WL, McKenna DR, et al: Toxic effect of chemotherapy dosing using actual body weight in obese versus normal-weight patients: A systematic review and meta-analysis. Ann Oncol 24:2952-2962, 2013

74. Grivennikov SI, Greten FR, Karin M: Immunity, inflammation, and cancer. Cell 140:883-899, 2010

75. Khandekar MJ, Cohen P, Spiegelman BM: Molecular mechanisms of cancer development in obesity. Nat Rev Cancer 11:886-895, 2011

76. Nguyen PL, Ma J, Chavarro JE, et al: Fatty acid synthase polymorphisms, tumor expression, body mass index, prostate cancer risk, and survival. J Clin Oncol 28:3958-3964, 2010

77. Ligibel JA, Alfano CM, Courneya KS, et al: American Society of Clinical Oncology position statement on obesity and cancer. J Clin Oncol 32:3568-3574, 2014

78. Rock CL, Doyle C, Demark-Wahnefried W, et al: Nutrition and physical activity guidelines for cancer survivors. CA Cancer J Clin 62:243-274, 2012

79. Schmitz KH, Courneya KS, Matthews C, et al: American College of Sports Medicine roundtable on exercise guidelines for cancer survivors. Med Sci Sports Exerc 42:1409-1426, 2010

AUTHORS' DISCLOSURES OF POTENTIAL CONFLICTS OF INTEREST

Obesity and Other Cancers

The following represents disclosure information provided by authors of this manuscript. All relationships are considered compensated. Relationships are self-held unless noted. I = Immediate Family Member, Inst = My Institution. Relationships may not relate to the subject matter of this manuscript. For more information about ASCO's conflict of interest policy, please refer to www.asco.org/rwc or jco.ascopubs.org/site/ifc.

Lin Yang
No relationship to disclose

Bettina F. Drake
No relationship to disclose

Graham A. Colditz
No relationship to disclose