

ERCIM



NEWS

[www.ercim.eu](http://www.ercim.eu)

Special theme:

# Machine Learning

Also in this issue:

Research and Society:

**Open Access**  
**Open Science**

*ERCIM News is the magazine of ERCIM. Published quarterly, it reports on joint actions of the ERCIM partners, and aims to reflect the contribution made by ERCIM to the European Community in Information Technology and Applied Mathematics. Through short articles and news items, it provides a forum for the exchange of information between the institutes and also with the wider scientific community. This issue has a circulation of about 6,000 printed copies and is also available online.*

ERCIM News is published by ERCIM EEIG  
BP 93, F-06902 Sophia Antipolis Cedex, France  
Tel: +33 4 9238 5010, E-mail: [contact@ercim.eu](mailto:contact@ercim.eu)  
Director: Jérôme Chailloux, ISSN 0926-4981

#### Contributions

Contributions should be submitted to the local editor of your country

#### Copyright notice

All authors, as identified in each article, retain copyright of their work. ERCIM News is licensed under a Creative Commons Attribution 4.0 International License (CC-BY).

#### Advertising

For current advertising rates and conditions, see <http://ercim-news.ercim.eu/> or contact [peter.kunz@ercim.eu](mailto:peter.kunz@ercim.eu)

#### ERCIM News online edition

<http://ercim-news.ercim.eu/>

#### Next issue

January 2017, Special theme: Computational Imaging

#### Subscription

Subscribe to ERCIM News by sending an email to [en-subscriptions@ercim.eu](mailto:en-subscriptions@ercim.eu) or by filling out the form at the ERCIM News website: <http://ercim-news.ercim.eu/>

#### Editorial Board:

Central editor:

Peter Kunz, ERCIM office ([peter.kunz@ercim.eu](mailto:peter.kunz@ercim.eu))

Local Editors:

Austria: Erwin Schoitsch ([erwin.schoitsch@ait.ac.at](mailto:erwin.schoitsch@ait.ac.at))

Belgium: Benoît Michel ([benoit.michel@uclouvain.be](mailto:benoit.michel@uclouvain.be))

Cyprus: Ioannis Krikidis ([krikidis.ioannis@ucy.ac.cy](mailto:krikidis.ioannis@ucy.ac.cy))

Czech Republic: Michal Haindl ([haindl@utia.cas.cz](mailto:haindl@utia.cas.cz))

France: Steve Kremer ([steve.kremer@inria.fr](mailto:steve.kremer@inria.fr))

Germany: Michael Krapp

([michael.krapp@scai.fraunhofer.de](mailto:michael.krapp@scai.fraunhofer.de))

Greece: Eleni Orphanoudakis ([eleni@ics.forth.gr](mailto:eleni@ics.forth.gr)), Artemios

Voyiatzis ([bogart@isi.gr](mailto:bogart@isi.gr))

Hungary: Andras Benczur ([benczur@info.ilab.sztaki.hu](mailto:benczur@info.ilab.sztaki.hu))

Italy: Carol Peters ([carol.peters@isti.cnr.it](mailto:carol.peters@isti.cnr.it))

Luxembourg: Thomas Tamisier ([thomas.tamisier@list.lu](mailto:thomas.tamisier@list.lu))

Norway: Poul Heegaard ([poul.heegaard@item.ntnu.no](mailto:poul.heegaard@item.ntnu.no))

Poland: Hung Son Nguyen ([son@mimuw.edu.pl](mailto:son@mimuw.edu.pl))

Portugal: José Borbinha, Technical University of Lisbon

([jl@ist.utl.pt](mailto:jl@ist.utl.pt))

Spain: Silvia Abrahão ([sabrahao@dsic.upv.es](mailto:sabrahao@dsic.upv.es))

Sweden: Kersti Hedman ([kersti@sics.se](mailto:kersti@sics.se))

Switzerland: Harry Rudin ([hrudin@smile.ch](mailto:hrudin@smile.ch))

The Netherlands: Annette Kik ([Annette.Kik@cwi.nl](mailto:Annette.Kik@cwi.nl))

W3C: Marie-Claire Forgue ([mcf@w3.org](mailto:mcf@w3.org))

## RESEARCH AND SOCIETY

The section “Research and Society” on “Open Access – Open Science” has been coordinated by Laurent Romary (Inria)

- 5 **Open Science: Taking Our Destiny into Our Own Hands**  
by Laurent Romary (Inria)
- 6 **ERCIM Goes to Open Access**  
by Jos Baeten (CWI) and Claude Kirchner (Inria)
- 7 **Will Europe Liberate Knowledge through Content Mining?**  
by Peter Murray-Rust (University of Cambridge)
- 9 **Roads to Open Access: The Good, the Bad and the Ugly**  
by Karim Ramdani (Inria)
- 10 **Open-Access Repositories and the Open Science Challenge**  
by Leonardo Candela, Paolo Manghi, and Donatella Castelli (ISTI-CNR)
- 11 **LIPICs – an Open-Access Series for International Conference Proceedings**  
by Marc Herbstritt (Schloss Dagstuhl – Leibniz-Zentrum für Informatik) and Wolfgang Thomas (RWTH Aachen University)
- 13 **Scientific Data and Preservation – Policy Issues for the Long-term Record**  
by Vera Sarkol (CWI)
- 14 **Mathematics in Open Access – MathOA**  
by Johan Rooryck and Saskia de Vries

## SPECIAL THEME

The special theme section “Machine Learnig” has been coordinated by Sander Bohte (CWI) and Hung Son Nguyen (University of Warsaw)

Introduction to the Special Theme

- 16 **Modern Machine Learning: More with Less, Cheaper and Better**  
by Sander Bohte (CWI) and Hung Son Nguyen (University of Warsaw)

More with less

- 18 **Micro-Data Learning: The Other End of the Spectrum**  
by Jean-Baptiste Mouret (Inria)
  - 19 **Making Learning Physical: Machine Intelligence and Quantum Resources**  
by Peter Wittek (ICFO-The Institute of Photonic Sciences and University of Borås)
  - 20 **Marrying Graphical Models with Deep Learning**  
by Max Welling (University of Amsterdam)
  - 22 **Privacy Aware Machine Learning and the “Right to be Forgotten”**  
by Bernd Malle, Peter Kieseberg (SBA Research), Sebastian Schrittwieser (JRC TARGET, St. Poelten University of Applied Sciences), and Andreas Holzinger (Graz University of Technology)
  - 24 **Robust and Adaptive Methods for Sequential Decision Making**  
by Wouter M. Koolen (CWI)
- Research
- 25 **Neural Random Access Machines**  
by Karol Kurach (University of Warsaw and Google), Marcin Andrychowicz and Ilya Sutskever (OpenAI (work done while at Google))
  - 26 **Mining Similarities and Concepts at Scale**  
by Olof Görnerup and Theodore Vasiloudis (SICS)
  - 28 **Fast Traversal of Large Ensembles of Regression Trees**  
by Claudio Lucchese, Franco Maria Nardini, Raffaele Perego, Nicola Tonello (ISTI-CNR), Salvatore Orlando (University of Venice) and Rossano Venturini (University of Pisa)

Massive data processing

**29 Optimising Deep Learning for Infinite Applications in Text Analytics**

by Mark Cieliebak (Zurich University of Applied Sciences)

**31 Towards Streamlined Big Data Analytics**

by András A. Benczúr, Róbert Pálovics (MTA SZTAKI), Márton Balassi (Cloudera), Volker Markl, Tilmann Rabl, Juan Soto (DFKI), Björn Hovstadius, Jim Dowling and Seif Haridi (SICS)

How does the brain do it?

**32 Autonomous Machine Learning**

by Frederic Alexandre (Inria)

**34 Curiosity and Intrinsic Motivation for Autonomous Machine Learning**

by Pierre-Yves Oudeyer, Manuel Lopes (Inria), Celeste Kidd (Univ. of Rochester) and Jacqueline Gottlieb (Univ. of Columbia)

Applications

**35 Applied Data Science: Using Machine Learning for Alarm Verification**

by Jan Stampfli and Kurt Stockinger (Zurich University of Applied Sciences)

**37 Towards Predictive Pharmacogenomics Models**

by George Potamias (FORTH)

**38 Optimisation System for Cutting Continuous Flat Glass**

by José Francisco García Cantos, Manuel Peinado, Miguel A. Salido and Federico Barber (AI2-UPV)

**40 Online Learning for Aggregating Forecasts in Renewable Energy Systems**

by Balázs Csanád Csáji, András Kovács and József Váncza (MTA SZTAKI)

**42 Bonaparte: Bayesian Networks to Give Victims back their Names**

by Bert Kappen and Wim Wiegerinck (University Nijmegen)

## RESEARCH AND INNOVATION

This section features news about research activities and innovative developments from European research institutes

**44 BASMATI – Cloud Brokerage Across Borders For Mobile Users And Applications**

by Patrizio Dazzi (ISTI-CNR)

**46 An Incident Management Tool for Cloud Provider Chains**

by Martin Gilje Jaatun, Christian Frøystad and Inger Anne Tøndel (SINTEF ICT)

**48 Predictive Modelling from Data Streams**

by Olivier Parisot and Benoît Otjacques (Luxembourg Institute of Science and Technology)

**49 Mandola: Monitoring and Detecting Online Hate Speech**

by Marios Dikaiakos, George Pallis (University of Cyprus) and Evangelos Markatos (FORTH)

**51 The BÆSE Testbed – Analytic Evaluation of IT Security Tools in Specified Network Environments**

by Markus Wurzenberger and Florian Skopik (AIT Austrian Institute of Technology)

**53 Behaviour-Based Security for Cyber-Physical Systems**

by Dimitrios Serpanos (University of Patras and ISI), Howard Shrobe (CSAIL/MIT) and Muhammad Taimoor Khan (University of Klagenfurt)

**54 The TISRIM-Telco Toolset – An IT Regulatory Framework to Support Security Compliance in the Telecommunications Sector**

by Nicolas Mayer, Jocelyn Aubert, Hervé Cholez, Eric Grandry and Eric Dubois

**56 Predicting the Extremely Low Frequency Magnetic Field Radiation Emitted from Laptops: A New Approach to Laptop Design**

by Darko Brodić, Dejan Tanikić (University of Belgrade), and Alessia Amelio (University of Calabria)

**57 Managing Security in Distributed Computing: Self-Protective Multi-Cloud Applications**

by Erkuden Rios (Tecnalia), Massimiliano Rak (Second University of Naples) and Samuel Olaiya Afolaranmi (Tampere University of Technology)

## EVENTS, IN BRIEF

Announcements

**59 VaMoS 2017: 11th International Workshop on Variability Modelling of Software-intensive Systems**

In Brief

**59 2016 Internet Defense Prize for Quantum-safe Cryptography**



# Fast Traversal of Large Ensembles of Regression Trees

by Claudio Lucchese, Franco Maria Nardini, Raffaele Perego, Nicola Tonellotto (ISTI-CNR), Salvatore Orlando (Ca' Foscari University of Venice) and Rossano Venturini (University of Pisa)

**The complexity of tree-based, machine-learned models and their widespread use in web-scale systems requires novel algorithmic solutions to make the models fast and scalable, both in the learning phase and in the real-world.**

Machine-learned models based on additive ensembles of regression trees have been shown to be very effective in several classification, regression, and ranking tasks. These ensemble models, generated by boosting meta-algorithms that iteratively learn and combine thousands of simple decision trees, are very demanding from a computational point of view. In fact, all the trees of the ensemble have to be traversed for each item to which the model is applied in order to compute their additive contribution to the final score.

This high computational cost becomes a challenging issue in the case of large-scale applications. Consider, for example, the problem of ranking query results in a web-scale information retrieval system: the time budget available to rank the possibly huge number of candidate results is limited due to the incoming rate of queries and user expectations of quality-of-service. On the other hand, effective and complex rankers with thousands of trees have to be exploited to return precise and accurate results [1].

To improve the efficiency of these systems, in collaboration with Tiscali Italia

S.p.A, we recently proposed QuickScorer (QS), a solution that remarkably improves the performance of the scoring process by dealing with features and characteristics of modern CPUs and memory hierarchies [2]. QS adopts a novel bit-vector representation of the tree-based model, and performs the traversal of the ensemble by means of simple logical bitwise operations. The traversal is not performed by QS one tree after another, as one would expect, but is instead interleaved, feature by feature, over the whole tree ensemble. Due to its cache-aware approach, both in terms of data layout and access patterns, and to a control flow that entails very low branch misprediction rates, the QS performance is impressive, resulting in speedups of up to 6.5x over state-of-the-art competitors.

An ensemble model includes thousands of binary decision trees, each composed of a set of internal nodes and a set of leaves. Each item to be scored is in turn represented by a real-valued vector  $x$  of features. As shown in Figure 1, the internal nodes of all the trees in the ensemble are associated with a Boolean test over a specific feature of the input

vector (e.g.,  $x[4] \leq \gamma_2$ ). Each leaf node stores the potential contribution of the specific tree to the final score of the item. The scoring process of each item requires the traversing of all the trees in the ensemble, starting at their root nodes, until a leaf node is reached, where the value of the prediction is considered. Once all the trees in the ensemble have been visited, the final score for the item is given by the sum of the partial contributions of all the trees.

One important result of QS is that to compute the final score, we only need to identify, in any order, all the internal nodes of the tree ensemble for which the Boolean tests fail, hereinafter false nodes. To perform this task efficiently, QS relies on a bit-vector representation of the trees. Each node is represented by a compact binary mask identifying the leaves of the current tree that are unreachable when the corresponding node test evaluates to false. Whenever a false node is found, the set of unreachable leaves, represented as a bit-vector, is updated through a logical AND bitwise operation. Eventually, the position of the leaf storing the correct contribution for each tree is identified. Moreover, in order to find all the false

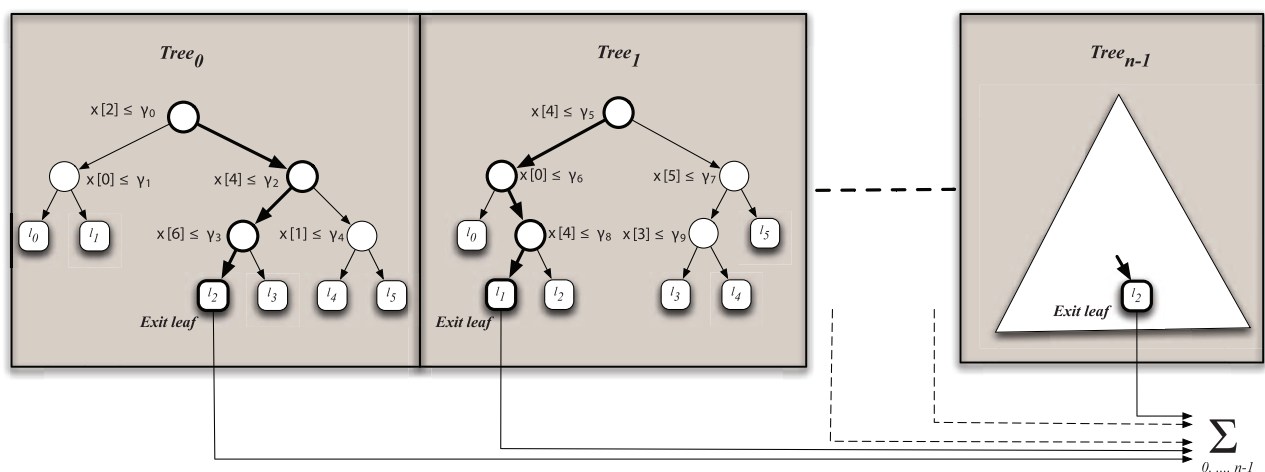


Figure 1: An ensemble of binary decision trees.

nodes for the scored item efficiently, QS processes the nodes of all the trees feature by feature. Specifically, for each feature  $x[i]$ , QS builds the list of all the nodes of the ensemble where  $x[i]$  is tested, and sorts this list in ascending order of the associated threshold  $\gamma k$ . During the scoring process for feature  $x[i]$ , as soon as the first test in the list evaluating to true is encountered, i.e.,  $x[i] \leq \gamma k$ , the subsequent tests also evaluate to true, and their evaluation can be safely skipped and the next feature  $x[i+1]$  considered.

This organisation allows QS to actually visit a consistently lower number of nodes than in traditional methods, which recursively visit the small and unbalanced trees of the ensemble from the root to the exit leaf. In addition, QS exploits only linear arrays to store the tree ensemble and mostly performs cache-friendly access patterns to these data structures.

Considering that in most application scenarios the same tree-based model is applied to a multitude of items, we recently introduced further optimisations in QS. In particular, we introduced vQS [3], a parallelised version of QS that exploits the SIMD capabilities of mainstream CPUs to score multiple items in parallel. Streaming SIMD Extensions (SSE) and Advanced Vector Extensions (AVX) are sets of instructions exploiting wide registers of 128 and 256 bits that allow parallel operations to be performed on simple data types. Using SSE and AVX, vQS can process up to eight items in parallel, resulting in a further performance improvement up to a factor of 2.4x over QS. In the same line of research we are finalising the porting of QS to GPUs, which, preliminary tests indicate, allows impressive speedups to be achieved.

More information on QS and vQS can be found in [2] and [3].

#### References:

- [1] G. Capannini, et al.: “Quality versus efficiency in document scoring with learning-to-rank models”, *Information Processing & Management*, Elsevier, 2016, <http://dx.doi.org/10.1016/j.ipm.2016.05.004>.
- [2] C. Lucchese et al.: “QuickScorer: A Fast Algorithm to Rank Documents with Additive Ensembles of Regression Trees”, *ACM SIGIR 2015*: 73-82 [best paper award].
- [3] Cl. Lucchese, et al.: “Exploiting CPU SIMD Extensions to Speed-up Document Scoring with Tree Ensembles”, *ACM SIGIR 2016*: 833-836.

#### Please contact:

Raffaele Perego  
ISTI-CNR, Pisa, Italy  
+39 (0)50 3152993  
[raffaele.perego@isti.cnr.it](mailto:raffaele.perego@isti.cnr.it)

## Optimising Deep Learning for Infinite Applications in Text Analytics

by Mark Cieliebak (Zurich University of Applied Sciences)

**Deep Neural Networks (DNN) can achieve excellent results in text analytics tasks such as sentiment analysis, topic detection and entity extraction. In many cases they even come close to human performance. To achieve this, however, they are highly-optimised for one specific task, and a huge amount of human effort is usually needed to design a DNN for a new task. With DeepText, we will develop a software pipeline that can solve arbitrary text analytics tasks with DNNs with minimal human input.**

Assume you want to build a software for automatic sentiment analysis: given a text such as a Twitter message, the tool should decide whether the text is positive, negative, or neutral. Until recently, typical solutions used a feature-based approach with classical machine learning algorithms (e.g., SVMs). Typical features were number of positive/negative words, n-grams, text length, negation words, part-of-speech tags etc. Over the last two decades a huge amount of research has been invested in designing and optimising these features, and new features had to be developed for each new task.

With the advent of deep learning, the situation has changed: now the computer is able to learn relevant features from the texts by itself, given enough

training data. Solving a task like sentiment analysis now requires three major steps: define the architecture of the deep neural network; aggregate enough training data (labelled and unlabelled); and train and optimise the parameters of the network.

For instance, Figure 1 shows the architecture of a system that won Task 4 of SemEval 2016, an international competition for sentiment analysis on Twitter [1]. This system uses a combination of established techniques in deep learning: word embedding and convolutional neural networks. Its success is primarily based on three factors: a proper architecture, a huge amount of training data (literally billions of tweets), and a huge amount of computational power to optimise its parameters. Live demos of various

deep learning technologies are available at [2].

#### Goal of DeepText

In DeepText, we will automate the three steps above as far as possible. The ultimate goal is a software pipeline that works as follows (see Figure 2):

1. The user uploads his or her training data in a standard format. The data can consist of unlabelled texts (for pre-training) and labelled texts, and the labels implicitly define the task to solve.
2. The system defines several DNNs to solve the task. Here, different fundamental architectures will be used, such as convolutional or recurrent neural networks.
3. The system then trains these DNNs and optimises their parameters.