Distributed Data Analytics using RapidMiner and BOINC

Nico Schlitter
Karlsruhe Institute of Technology
Steinbuch Centre for Computing
Nico.Schlitter@kit.edu

Jörg Lässig
University of Applied Sciences Zittau/Görlitz
Enterprise Application Development Group
Joerg.Laessig@hszg.de

Abstract

RapidMiner is an open source environment for machine learning and data analytics. It is intensively used for academic purposes at universities as well as for industrial or commercial applications. The BOINC framework also attracted attention as it provides the ability to easily setup a distributed computing environment. This article addresses the joint usage of RapidMiner and BOINC. We describe the integration of both tools and present some of the research accomplishments of the distributedDataMining.org project.

1 Introduction

The term big data refers to the challenge of capturing, storing or processing of huge data sets. Due to the amount of data, traditional data management and data analysis approaches are no longer feasible. High performance computing environments are traditionally used to overcome the problem. However, applying state-of-the-art techniques like MapReduce [2] in high performance grid computing infrastructures may exceed the financial abilities of most researchers. Therefore, easy to use infrastructures are needed in order to supply scientists with cost efficient computing resources for data analytics.

In this article, we introduce a distributed computing project called *distributedDataMining.org* which supports scientists from different research areas by providing computing power for data analysis purposes. We describe the

usage of the BOINC framework and the distribution of research related computing tasks to thousands of heterogeneous computing nodes located all over the world. Each of these nodes uses the BOINC client to pull computing tasks and data packages from a central server. Then, the BOINC client starts an instance of the RapidMiner framework in order to process the data mining tasks. The data mining results are sent back to a central project server, which gathers the information and provides it to researchers for further analysis.

The remainder of this article is organized as follows. Section 2 sketches the machine learning environment RapidMiner and the distributed computing framework BOINC. Both tools are used to apply distributed data analysis tasks within the *distributedDataMining.org* project. This distributed computing project is introduced in Section 3. An overview of the research cooperations in the field of Social Network Analysis, Time Series Analysis and Biological Data Analysis is given in Section 4. Finally, we conclude in Section 5.

2 Tools

2.1 RapidMiner

RapidMiner [8] is an environment for machine learning, data mining, text mining, predictive analytics, and business analytics. The RapidMiner project was started in 2001 by Ralf Klinkenberg, Ingo Mierswa, and Simon Fischer at the Artificial Intelligence Group of Katharina Morik at the Dortmund University of Technology.

In 2007, the project formally known as YALE was renamed and published as RapidMiner version 4.0. Since then, the software is hosted by SourceForge and is offered free of charge as a Community Edition released under the GNU AGPL. There is also an Enterprise Edition offered under a commercial license for integration into closed-source projects.

The software is written in Java and runs so called processes. A process is basically an XML-File generated by the user and contains a sequence of tasks which are represented by operators. More than 500 operators are already included in the software. Their functionality covers the main aspects of data analysis such as data loading and transformation, data preprocessing and visualization, modelling and model evaluation. By combining these operators, basic machine learning tasks such as data mining, text mining, time series analysis and forecasting, web mining as well as sentiment analysis and opinion mining can be performed. The software also provides multiple methods for visualizing high dimensional data sets. Since RapidMiner is written in Java it is platform independent and can be easily combined with other software tools. Doing so, the well known WEKA framework [7] was completely integrated into RapidMiner. In addition, RapidMiner provides a magnificent

plug-in mechanism, which can be used to easily expanded the functionality of the core software.

Since 2007, RapidMiner has been heavily extended and became one of most important data mining and data analytic tools. It is intensively used in introductory courses and academic purposes at universities all over the world. RapidMiner is also used for industrial purposes by many companies and consultants for different applications.

2.2 BOINC

The *Berkeley Open Infrastructure for Network Computing* (BOINC) is a software framework for distributed and grid computing [1]. It was originally developed at the University of California at Berkeley for the SETI@HOME program which was founded to analyze radio signals, searching for signs of extra terrestrial intelligence. The BOINC project started in 2002 by releasing its first version under the terms of the GNU LGPL.

One main objective of the BOINC development team is the separation of project management and research related tasks. The framework supports researches by providing the necessary infrastructure to distribute computational intensive research tasks to several computers which are running the BOINC client. The BOINC client downloads tasks for one or more research projects, processes them and sends the results back to the project server. Thus, the scientists can rather focus on developing analysis algorithms than taking care of distributing data and software to different locations by themselves.

Volunteers, which want to provide their computing resources to research purposes, download the BOINC client and connect it to one of the many existing research projects powered by BOINC. The client then autonomously downloads all necessary data and analysis programs and starts contributing to the project goals. Based on the amount of computing time spend, the volunteers get rewarded by credit points. Even though these credit points only possess an immaterial value and can not be used to buy anything, they are subject to competitive behaviour of the volunteers. The earned credits are recognized as measurement for their willingness to contribute to scientific progress and therefore motivate volunteers to participate in BOINC projects.

In March 2013, the computing power of about 400 thousand computers was spent by volunteers to contribute to BOINC powered research projects. This leads to an average of 9 PetaFLOPS and outperforms most of the high performance compute clusters of the top 500 list.

2.3 Drupal

Drupal is a free and open-source content management framework written in PHP and distributed under the terms of the GNU LGPL. The standard re-

lease of Drupal contains basic features common to content management systems. These include delivery of dynamic and static websites, user account registration and maintenance, menu management, RSS feeds, and system administration. The Drupal system can be easily customized in behaviour and appearance. The core functionality can be extended by addons, which allow to add new features. The Drupal theming mechanism provides the possibility to integrate different themes in order the change the layout and design of the website.

2.4 Tool integration

We were looking for a system that allows us to perform independent data analysis tasks using a cost-efficient but high performance computing infrastructure. We decided to combine the data analysis functionality of RapidMiner and BOINC's capability of job distribution. The advantages are obvious - both frameworks are free and open source software under AGPL and LGPL. The actual computing resources are provided by enthusiastic volunteers which gracefully support our research challenges by providing free of charge computing power. Our investment was limited to the renting and housing of an dedicated server and a high-throughput network connectivity. In addition, we spend approximately six person-months to extend and customize the available software and to develop an automatic data management workflow.

We implemented a module which allows Drupal to interact with the BOINC framework. Thus, we were able to replace the standard web appearance of BOINC and benefit from Drupal features. We extended the module by adding charts, user profiles, a forum and all kind of statistical information in order to adjust to needs of our project members. Doing so, we combined a state-of-the-art content management system and the BOINC framework in order to supply our members with up to date news.

In our setup, the RapidMiner environment is used to carry out independent data analysis tasks, which are distributed to and performed on an armada of volunteer computers. In this situation, we were facing a highly heterogeneous infrastructure in terms of operating systems, amount of usable memory (RAM and HDD) as well as types and number of available CPUs. Fortunately, the BOINC frameworks provides powerful tools to cope with this situation.

In order to benefit from the immense computational power provided by the BOINC framework and the committed volunteers, it is necessary to divide the overall data analysis process into a large number of small independent analysis tasks. Each of these tasks contains a portion of the data that needs to be analysed and a description of the data analysis process which is to be applied. In addition, some meta information is assigned to each task, such as the reference to the application that performs the actual data analysis and its estimated run time.

BOINC provides a C++ library which enables a given scientific application to communicate with the BOINC client. This communication is essential, since the BOINC client needs to be able to start, suspend, resume or stop the application. In addition, the application has to provide the progress of its computation during run time.

Unfortunately, there is currently no Java API available, which allows a developer to add this functionality to a certain java application. Therefore, we had to develop an C++ application for Windows and Linux environments, that worked as a wrapper for the java-based RapidMiner. Our wrapper application receives the communication signals from the BOINC client and controls the RapidMiner software accordingly. The start signal leads to the execution of a Java Runtime Environment (JRE) which then starts a RapidMiner instance. Once started, suspending and resuming of execution can be easily achieved by using the appropriate OS-specific system calls. During suspension the RAM is preserved and after resuming the application continues its execution seamlessly. The stop signal immediately shuts down the JRE including the RapidMiner instance. Furthermore, an application dependent checkpoint file is written which stores the computing progress reached so far. This check point information is used when the start signal is sent by the BOINC client. The wrapper interprets the checkpoint information and manipulates the data and process files in order to restart the data mining task at a certain point.

These manipulations are highly application dependent and can not be generalized. Each RapidMiner analysis process needs a specific wrapper application which controls the checkpoint writing, reading and interpretation. The implementation of these specific wrappers and checkpointing mechanisms is time consuming and a huge conceptual effort for each analytical process.

The effort gets minimized if the analysis process contains of steps that are frequently repeated. An example would be a time series analysis task, which uses a sliding window approach to train a classification model and applies it on the time window in order to predict the next time series value. The necessary steps are the same for each time window and repeated as often as there are time windows. The checkpointing could work in a way, that the predictions for each time window are added to an overall result file. When the wrapper restarts the analysis process it first checks the number of predictions, that are already in the result file and then adapts the data file by removing the first time windows which are already processed.

The described approach can be used for each analysis or optimization process which contains multiple repeated sub-processes. Other examples which might be dealt with in a similar manner are parameter optimization tasks using random search, linear search or genetic algorithms.

The problem of checkpointing can be avoided if the run time of a single analysis process is quite short. In such a case, checkpointing can be waived

since the restart of an yet unfinished task would only lead to loosing a small amount of CPU time.

3 The Distributed Data Mining Project

The distributedDataMining project is a scientific computing project that provides the computational power of internet-connected computers to its scientific partners in order to perform research in the various fields of Simulation, Data Analysis and Machine Learning. Since 2008, the project uses the BOINC framework for the distribution of data analysis tasks which are then performed by the RapidMiner environment on computers of enthusiastic and committed volunteers. The project became available to the public in March 2010.

The project's goal is to allow our research partners to make use of the enormous processing power of personal computers around the world. Thereby, the computational power spent by the project members is used to support the research of our scientific partners. Figure 1 shows the number of volunteers which spent the computational power of their computers for research purposes over the time period of three years. The number of participating computers is shown in Figure 2.

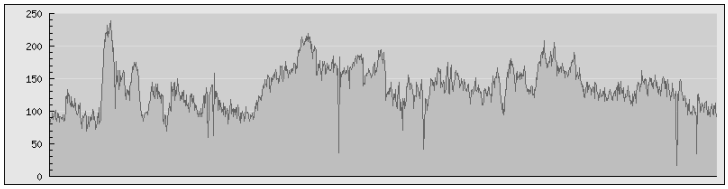

Figure 1: Number of contributing volunteers (06/2010 - 06/2013)

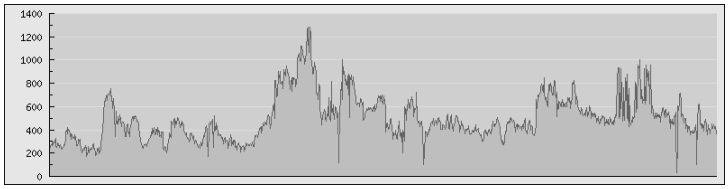

Figure 2: Number of contributing hosts (06/2010 - 06/2013)

The computing power as visualized in the figures was used to process different data analysis tasks from various research areas. The research areas as well as the related applications and case studies are briefly introduced in the following section.

4 Applications

4.1 Social Network Analysis

In recent years more and more social network platforms have been established. Some prominent examples are Facebook, Xing and LinkedIn. The platform providers collect a huge amount of data for each of their users. Besides personal information such as age, hobbies or professional career, also relations between users are specified. Consequently, each platform can be represented as a network of users where each user can be characterized with specific properties. The links between users may have different meanings. A very common meaning is that two users are connected if they know each other.

The data in social networks contains valuable information. In the following, we present some results, which were obtained by using the DenGraph algorithm. DenGraph was implemented for the RapidMiner and distributively processed using the BOINC framework.

4.1.1 The DenGraph algorithm

Inspired by the algorithm DBSCAN [3] for spatial data, Falkowski et al. propose the density based graph clustering algorithm DenGraph [5]. The intention of DenGraph is to cluster similar nodes in a graph into communities. The density-based approach applies a local cluster criterion. Clusters are regarded as regions in the graph in which the nodes are dense, and which are separated by regions of low node density.

To allow for tracking and analyzing the temporal dynamics of Social Network Communities, DenGraph-I [6] is designed as an incremental procedure: The clustering is updated incrementally based on the changes that are observed in the graph structure from one interval to another. These changes may evoke one of the following clustering updates: creation of a new cluster, removal of a cluster, absorption of a new cluster member, reduction of a cluster member, merge of two or mores clusters and split of a cluster into two or more clusters.

In social networking sites it is often observable that members belong to more than one community. So far, if a member is close to more than one community, it is assigned to the cluster which is discovered first. In this case, the clustering result is not deterministic but depends on the order in which the nodes are visited. To overcome this problem we propose DenGraph-IO that extends the existing algorithm to handle overlapping clusters. By this, we also achieve a more realistic clustering as individuals can be members in different communities now.

In 2011, we proposed DenGraph-HO in order to fulfill the special needs of social network analysts [13, 14]. In most cases, the visual inspection of a

network is the first step of the analytical process and helps to determine the basic graph characteristics and further actions. DenGraph-HO supports this early stage by providing a quick visual analysis of the network structure. It provides the ability of zooming into network clusterings and has proven its usefulness for our practical work.

The algorithm's approach differs from traditional hierarchical clustering methods in that DenGraph-HO is a non-partional clustering algorithm. We consider the fact that not all nodes are necessarily members of clusters. In addition, the proposed hierarchy is not strictly built up by the classic divisive or agglomerative approach that is known from literature. We generalize these methods and propose a top-down approach and a bottom-up approach by extending the hierarchy paradigms. The proposed hierarchy supports superordinate clusters that contain subclusters.

Each level of the hierarchy represents a clustering that fulfills the original DenGraph paradigms. The levels, respectively the clusterings, differ in the density that is required to form a cluster. While lower level clusterings aggregate nodes with a lower similarity, higher level clusterings require a higher similarity between nodes. The efficiency of our algorithm is based on this iterative sequence of cluster adaptations instead of a complete new clustering.

4.1.2 Last.fm analysis

In 2008, we applied the DenGraph-IO on a music data set to analyse the music listen behaviour of users on the Last.fm platform [4, 12]. Last.fm is a social networking platform established in 2002. The platform has over 20 million users on the site every month, which are based in more than 200 countries. After a user signs up, Last.fm records - among others - all artists a user listens to, aggregates this information over seven days and provides lists of the most listened artists for each week over the lifetime of a user. We use this information to build a user's profile by extracting the genres of the most listened artists. The artist's genre is determined by the tags that the community members use to characterize the artist. We represent each user as node in a graph and connect users with an edge, if their profile similarity reaches a predefined threshold. The similarity is determined by calculating the distance between pairs of genre vectors using the cosine similarity measure.

For our study, we randomly chose approximately 600,000 users and obtained their weekly artists charts over a period of 167 weeks (September 2005 to November 2008). Since many users are not active on a regular basis, we chose randomly 2,000 users from this set who were active in at least 80% of all periods. We applied DenGraph-IO on the resulting graph to detect and observe the evolution of clusters during the observation period of 115 weeks. The aim was to see, whether the proposed clustering technique detects meaningful communities and evolutions.

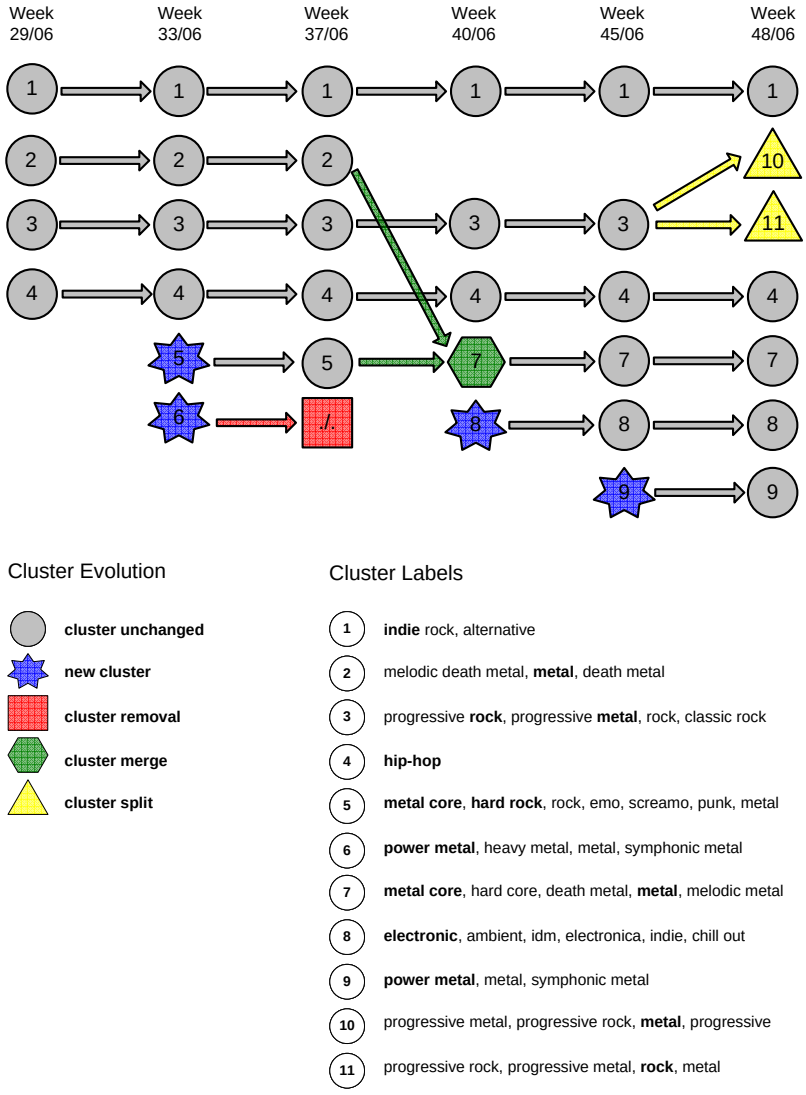

Figure 3: DenGraph-HO: Last.fm hierarchy

Figure 3 shows the evolution of clusters found by the DenGraph-IO algorithm. At first, only four clusters were found. These clusters represent the music genres indy, metal, rock and hip hop. The algorithm then tracks these clusters and detects structural changes. Four weeks later two additional clus-

ters were found. One of them disappears in the next step. In week 40/2006, the clusters which represent the metal genre got merged in one bigger cluster. In week 48/2006 the rock cluster is split in two subclusters.

In 2011, we applied the DenGraph-HO algorithm to the Last.fm graph consisting of 1,209 nodes and 12,612 edges. The resulting clusters form groups of users that have similar music listening preferences. By calculating labels the clusters get a semantic meaning based on the music preferences of its members. Figure 4 shows the resulting hierarchy of clusters which represent music genres. The underlying graph and the discovered clusters are shown in Figure 5. For the sake of clarity the graph edges are not drawn. In both the cluster hierarchy and the graph, clusters with similar labels are located closely in the graph and in the hierarchy.

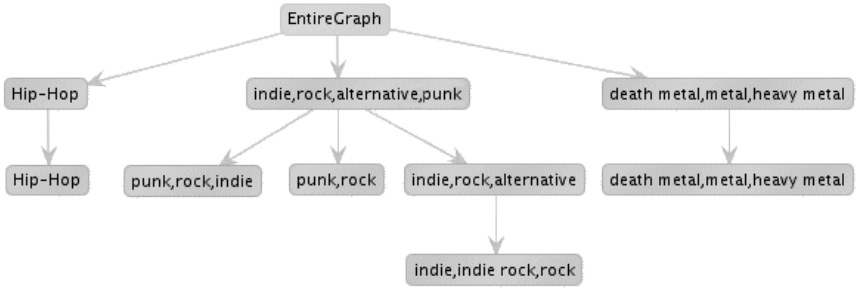

Figure 4: DenGraph-HO: Last.fm hierarchy

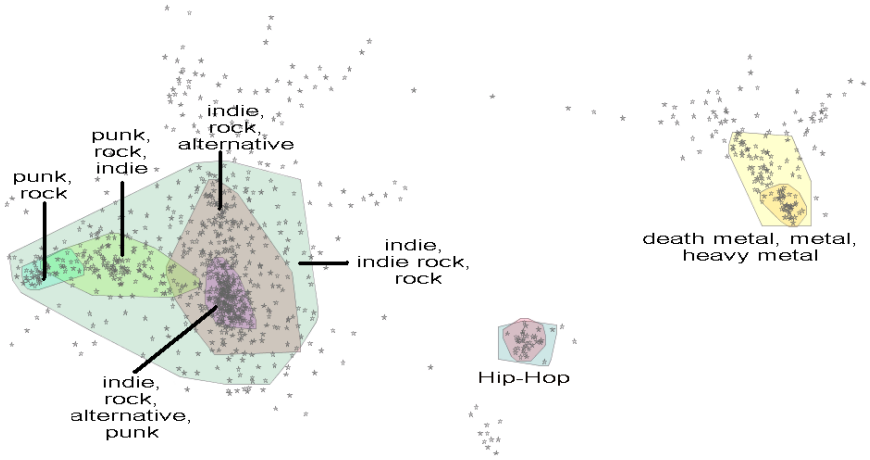

Figure 5: DenGraph-HO: Last.fm Graph and Clustering

4.2 Time Series Analysis

A Time Series is an ordered sequence of data points, which are typically measured at uniform time intervals. The research area called Time Series Analysis comprises methods for analyzing time series data in order to extract meaningful statistics, rules and patterns. Later on these rules and patterns might be used to build forecasting models that are able to predict future developments. In case one wants to predict future trend directions (e.g. up/down) a classification problem has to be solved. If we try to forecast future time series data points, the relevant data mining technique is called regression.

Within the distributedDataMining.org project, we use different machine learning algorithms to discover and extract valuable patterns which are embedded in financial time series. These patterns are subsequently used to built forecasting models which should be able to predict future developments. The algorithms we use are integrated in the open source data mining framework RapidMiner. For classification problems we applied Decision Trees, k-nearest Neighbours, Support Vector Machines and Neural Networks. Furthermore, Linear Regression, LeastMedSquare Regression and Logistic Base Regression are used to build regression models.

In 2008, we published some studies focusing on stock price prediction [9, 10, 11]. The proposed methodology lead to prediction models which reach nearly always a positive profit gain. We succeeded in generating models that outperform the general market. Furthermore, our results showed some interesting news. The amount of historical data had a much higher impact on the prediction quality than expected. This fact has not been taken into account in other studies before.

4.3 Biological Data Analysis

4.3.1 Laryngeal high-speed video classification

For the clinical diagnosis of pathological conditions of the human body a variety of sophisticated examination techniques are employed these days. Most of these approaches yield vast amounts of images and measurement data with high spatial and/or temporal resolutions, e.g. MRI, CT, and Ultrasound. In order to reliably evaluate these data for diagnostic purposes, a certain extent of subjective experience is required on the part of the physician. Due to different reasons, in usual clinical time frames the amount of time available for analyzing and interpreting the acquired data is limited. As a result, diagnostic failure may occur, which can have serious consequences for the affected patient. By means of combined image processing and data analysis approaches this crucial diagnostic process can be objectified and automated. Thus, Computer-Aided Diagnosis systems can be provided to the physician, facilitating her/his clinical

decision and yielding more reliable identification of pathological alterations.

One particular field of interest within this medical context is the automatic identification of voice disorders, resulting in perceivable hoarseness. Commonly, for this purpose audio recordings of the acoustical voice signal are analyzed with specialized software quantifying the amount of perturbation (noise) in the signal. However, this type of acoustical analysis does not allow for the clear assignment of certain clinical pictures to a distinct set of perturbation parameters. A more revealing approach for voice diagnosis consists in endoscopic examination of the sound-producing vocal folds in the larynx by means of digital high-speed cameras. These cameras are capable of recording the laryngeal movements at a frame rate of several thousand images per second, and thus, allowing for conclusive real-time analysis. However, the task of manually analyzing the resulting high-speed videos is time-consuming and error-prone. Through automated feature extraction from the recordings and subsequent machine learning analysis, laryngeal movement patterns can be quantitatively captured and automatically classified according to different diagnostic classes (e.g. organic and functional dysphonia). By means of the distributedDataMining infrastructure, we evaluated a large number of machine learning paradigms (e.g. Support Vector Machines, Artificial Neural Networks) and corresponding parameter optimization strategies (e.g. Grid search, Evolution strategy, Genetic algorithms). This preliminary evaluation step allowed us to identify certain learning schemes and parameters which are particularly suited for the considered clinical classification task. Details on the proposed methodology and the obtained classification results can be found in [15, 16, 17, 18]

4.3.2 Multi-Agent Simulation of Evolution

In this case study, we investigate the biological phenomenon of aposematism (also referred to as warning coloration). This term describes the evolutionary strategy of certain animal species to indicate their unpalatability/toxicity to potential predators by developing skin colors and patterns that can be easily perceived by them. Prominent examples of toxic animals with distinct warning coloration are poison dart frogs, coral snakes and fire salamanders.

The evolution of aposematism has intrigued many biologists, because at first glance, an evolutionary paradox seems to be underlying: Why would unpalatable prey animals acquire conspicuous warning coloration if this trait makes them more likely to be spotted and eaten by predators? Given that aposematism can be frequently observed in the animal world, the question arises how these warning signals could have evolved so many times despite their apparent evolutionary disadvantage. The paradox is even aggravated by the fact that in its initial stage of evolution, the proposed benefit of aposematic colors (i.e. making it easier for predators to learn the prey's unpalatability) cannot be present. Consequently, the evolution of aposematism has spurred

more than a century of scientific discussion and investigation; it has been addressed both experimentally and theoretically.

For tackling this interesting research challenge, we developed a distributed multi-agent model that simulates the dynamic interactions of predator and prey populations over time. By systematically testing different adaptation and learning strategies for the agents and exploring the parameter space of our simulation model using the computational power of the distributedDataMining.org project, we might be able to deepen the understanding of the aposematism phenomenon and the evolutionary paths leading to it.

5 Conclusion

In this article, we briefly described the distributedDataMining.org project which uses BOINC, Druva and RapidMiner for distributed data analytics. We gave an overview about the integration of the used tools and how they interact. In addition, we presented our data analysis results which were achieved in the research fields of Social Network Analysis, Time Series Analysis and Biological Data Analysis. The distributedDataMining.org project is up and running and offers its computational resources to interested researchers and their institutions.

References

[1] David P. Anderson. Boinc: A system for public-resource computing and storage. In *Proceedings of the 5th IEEE/ACM International Workshop on Grid Computing*, GRID '04, pages 4–10, Washington, DC, USA, 2004. IEEE Computer Society.

[2] Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. *Commun. ACM*, 51(1):107–113, January 2008.

[3] Martin Ester, Hans-Peter Kriegel, Jrg S, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

[4] Tanja Falkowski. *Community Analysis in Dynamic Social Networks*. Sierke Verlag, Goettingen, 12 2009.

[5] Tanja Falkowski, Anja Barth, and Myra Spiliopoulou. Dengraph: A density-based community detection algorithm. In *In Proc. of the 2007 IEEE / WIC / ACM International Conference on Web Intelligence,*, pages 112–115, 2007.

[6] Tanja Falkowski, Anja Barth, and Myra Spiliopoulou. Studying community dynamics with an incremental graph mining algorithm. In Izak Benbasat and Ali R. Montazemi, editors, *AMCIS*, pages 29–29. Association for Information Systems, 2008.

[7] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009.

[8] Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz, and Timm Euler. Yale: Rapid prototyping for complex data mining tasks. In Lyle Ungar, Mark Craven, Dimitrios Gunopulos, and Tina Eliassi-Rad, editors, *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 935–940, New York, NY, USA, August 2006. ACM.

[9] Manuel Möller and Nico Schlitter. *Analyse und Prognose ökonomischer Zeitreihen mit Support Vector Machines*, pages 189–201. Chemnitzer Informatik-Berichte. Techn. Univ. Chemnitz, Fak. für Informatik, Chemnitz, April 2008.

[10] Nico Schlitter. *Analyse und Prognose ökonomischer Zeitreihen: Neuronale Netze zur Aktienkursprognose*. VDM Verlag Dr. Müller, Saarbrücken, July 2008.

[11] Nico Schlitter. *A Case Study of Time Series Forecasting with Backpropagation Networks*, pages 203–217. Chemnitzer Informatik-Berichte. Techn. Univ. Chemnitz, Fak. für Informatik, Chemnitz, April 2008.

[12] Nico Schlitter and Tanja Falkowski. Mining the dynamics of music preferences from a social networking site. In *Proceedings of the 2009 International Conference on Advances in Social Network Analysis and Mining*, pages 243–248, Athens, 2009 2009. IEEE Computer Society, IEEE Computer Society.

[13] Nico Schlitter, Tanja Falkowski, and Jörg Lässig. Dengraph-ho: Density-based hierarchical community detection for explorative visual network analysis. In *Research and Development in Intelligent Systems XXVIII Incorporating Applications and Innovations in Intelligent Systems XIX Proceedings of AI-2011, the Thirty-first SGAI International Conference on Innovative Techniques and Applications of Artificial Int*, pages 283–296, Cambridge, 2011. Springer, Springer.

[14] Nico Schlitter and Jörg Lässig. Distributed privacy preserving classification based on local cluster identifiers. In *2012 IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, Liverpool, 2012. IEEE Computer Society, IEEE Computer Society.

[15] D. Voigt and U. Eysholdt. Identifying relevant analysis parameters for the classification of vocal fold dynamics. *J Acoust Soc Am*, 130(4):2550, 2011.

[16] Daniel Voigt. *Objective Analysis and Classification of Vocal Fold Dynamics from Laryngeal High-Speed Recordings*. Shaker Verlag GmbH, Aachen, 2010.

[17] Daniel Voigt, Michael Döllinger, Thomas Braunschweig, Anxiong Yang, Ulrich Eysholdt, and Jörg Lohscheller. Classification of functional voice disorders based on phonovibrograms. *Artificial Intelligence in Medicine*, 49(1):51–59, 2010.

[18] Daniel Voigt, Jörg Lohscheller, Michael Döllinger, Anxiong Yang, and Ulrich Eysholdt. Automatic diagnosis of vocal fold paresis by employing phonovibrogram features and machine learning methods. *Comput Methods Programs Biomed*, 99:275–288, 2010.