ORANGE: DATA MINING FRUITFUL AND FUN

Janez Demšar and Blaž Zupan
University of Ljubljana, Faculty of Computer and Information Science
Tržaška 25, 1000 Ljubljana, Slovenia
e-mail: (janez.demsar,blaz.zupan)@fri.uni-lj.si

ABSTRACT

Orange (http://orange.biolab.si) is a general-purpose machine learning and data mining tool. It features a multi-layer architecture suitable for different kinds of users, from inexperienced data mining beginners to programmers who prefer to access the tool through its scripting interface. In the paper we outline the history of Orange's development and present its current state, achievements and the future challenges.

1 INTRODUCTION

The history of general-purpose machine learning software tools is short but eventful. It all started with utilities that implemented a specific induction method and printed out the model in a textual form, to be then scrutinized and admired by the user. Such were implementations of tree and rule inducers C4.5 [1] and CN2[2] in the 1980s. Contender of C4.5 was Assistant Professional [3], but as C4.5 came free with a book, and Assistant Professional was pricy, C4.5 took off and became, in the last century, perhaps the most known machine learning program and, in the spirit of free software, a precursor of open-source toolboxes.

In the 1990s, machine learning community grew substantially and so did the number of different approaches and the desire to have a single toolbox where these could be implemented, used and tested against the same data sets. Packages like IND and MLC++[1] took off, each with set of command-line utilities that also implemented some testing schemes and could report on evaluation scores.

Having a command-line utility and only a textual interface is actually very cumbersome, if, imagine, a medical doctor asks you which patients actually went to some branch of the tree. Or what is their blood pressure? Command-line data analytics is no fun. Thus, systems with graphical user interfaces emerged. Not only systems with data plotting capabilities like R[2], but instead those where graphics was interactive and where a user could query on some graphical element to get the information of what data it represents. SGI's MineSet[4], for one, was a commercial product build on a top of MLC++ and implementing what was for 1990s an advanced exploratory graphical interface. There was one problem: the data analysis pipeline was fixed. Read the data, visualize the tree, explore it. Or show pie charts for naïve Bayesian Classifier.

People are in general inquisitive. That is, given the data, we like to dissect it, build some models, observe its parts, consider their specific data subsets and dissect it further. We like to construct the data analysis pipelines, not just use it. The ideas of visual programming, an interface where pipelines are created by linking pre-defined or even user-designed components, was in early 1990s available from Sun and SGI in packages for data visualization like Data Explorer. Similar idea in data mining took off by Clementine (then bought by SPSS and in 2009 renamed to SPSS Modeler). Open-source toolboxes followed; Weka[3], Knime[4], Yale (what is now a much redesigned RapidMiner[5]), and Orange[6]. Each building on their own favorite programming languages, assembling a different set of core components, and offering their own landmark interface for explorative data analysis.

The authors of this paper are great believers in component-based software. Along came Python, Qt for interoperability and the visual programming. Orange was born in mid 1990s, and is, along Weka and R, a data analysis toolbox with perhaps a longest (short) history. We still enjoy it, and continue to improve it.

2 HISTORY OF ORANGE

Development of Orange began in 1997 by the two authors of this paper, then members of the Artificial Intelligence Laboratory and now at Laboratory of Bioinformatics, at the University of Ljubljana.

Orange as a library of C++ components and command-line utilities. Orange was first conceived as a C++ library of machine learning algorithms and related procedures, like preprocessing, sampling and other data manipulation. We soon found out, though, that we seldom needed to write specialized applications that would require the use of C++ components. Instead, Orange was mostly used for data exploration in which different combinations of preprocessing and learning algorithms were tested and scored using cross validation. The components were packed into programs that could be used via command line interface. As this has soon proven limiting, we decided to provide a scripting interface to these components by exposing them to Python.

[1] http://www.sgi.com/tech/mlc
[2] http://www.r-project.org
[3] http://www.cs.waikato.ac.nz/ml/weka
[4] http://www.knime.org
[5] http://rapid-i.com
[6] http://orange.biolab.si

Orange as Python module. Python is a modern scripting language that was chosen for variety of reasons.

- It has a very clean and simple syntax that is easy to learn not only for a programmer but also for a beginner. (Python is, in fact, becoming the language of choice for basic courses in programming at many leading US universities, including CMU, MIT, Berkeley, Rice and Caltech.)
- Despite its simplicity, Python is an industry-strength language. For instance, Python is behind many of Google's technologies, which is also why Google is one of the major sponsors of Python's development.
- Since programming in Python is fast, it is very suitable for prototyping of new methods.
- Python allows for relatively easy extension with modules written in C or C++. For this reason, Python is occasionally dubbed a glue language due to its use for gluing libraries in C or Fortran. Lately, one seldom needs to implement specialized routines in low level languages due to availability of high quality libraries like numpy and scipy.

From 1999, Orange was used almost exclusively as a Python module. While the C++ core eventually rose to around 140,000 lines of code in C++, most developers have been adding to its Python modules and avoiding C++.

Transition to Python enabled several important developments. More and more of Orange's functionality is implemented in pure Python or by combining the fast functions provided by the Orange core using the glue code written in Python. Since the programs in Python are so readable, they enable collaboration of larger teams without the need to coordinate the development and establish a set of coding standards. The size of the group that develops the system has increased to 10-15 members, mostly from (today's) Laboratory of Bioinformatics. Most importantly, migration to Python simplified the development of the graphical user interface.

Orange Visual Programming. Our group has a tradition of collaboration with partners from other scientific and industrial areas, in particular from biomedicine. We wished to provide them with a data exploration tool where they could design their own data analysis pipelines without any scripting or Python programming [5]. Among a number of different Python libraries for GUIs we decided for Qt, which is a strong cross-platform library available under both GPL and commercial licenses and is behind products as different as Skype and KDE desktop.

Majority of current users now use Orange only through its graphical interface. It consists of a canvas onto which the users place pipeline components called widgets. Each widget offers some basic functionality, like reading the data, showing a data table, selecting features, either manually or based on some feature scoring, training predictors, cross-validating them and so forth. The user connects the widgets by communication channels. The basis strength and flexibility of Orange is in different ways in which the widgets can be combined into new schemata.

Special emphasis in development and design of widget was placed on data visualization and interactivity. For instance, a classification tree viewer allows the user to click a node in the tree. Doing so transmits the data samples that belong to the node to any widgets connected to the tree viewer widget. The user can thus construct a tree and then explore its content by observing, say, a data table with the data instances from interesting nodes, or, for example, by drawing scatter plots for data from different nodes of the tree.

3 ORANGE IN 2012

Currently, Orange is, together with Knime, perhaps one of the easiest-to-use data mining tools around. It can be run on OS X, Windows and Linux, and can also be parallelized on a grid.

The default installation includes a number of machine learning, preprocessing and data visualization algorithms. Opposed to, for instance, Weka, that offers everything there is in machine learning, the goal of Orange was to implement the most useful and commonly used techniques in a way that is flexible and user-friendly; the emphasis of the tool is on data exploration. For instance, the machine learning algorithms in the default installation are limited to naive Bayesian classifier, k nearest neighbors, induction of rules and trees, support vector machines, neural networks, linear and logistic regression, and ensemble methods. Most methods are, however, coupled with a visual representation that allows for exploration of the resulting module; the user can select a node in a classification tree or a rule and explore the training instances covered by them. Naive Bayesian classifier, logistic regression and linear SVM can be explored through nomograms that offer insight into importance of features and their individual values, and can also be used for explaining the model's predictions. Similar goes for unsupervised methods, such as association rules, multidimensional scaling, self-organizing maps and various types of clustering.

For a contrast from the intentionally limited assortment of machine learning methods, Orange has a rich collection of visualization methods: besides the common visualizations, like box plots, histograms and scatter plots, it contains a number of multivariate visualizations such as parallel coordinates, mosaic display, sieve diagram, survey plots and a number of data projection techniques, like multi-dimensional scaling, principal component analysis, RadViz, FreeViz and others. The user can interactively explore the visualizations or connect them to other widgets that send or receive the data from the visualization. Orange can also help the user in finding insightful visualizations by automatically ranking them by interestingness or by organizing them into a network of visualizations.

Orange also contains powerful widgets for visualization and exploration of networks, again with focus on interaction and flexibility.

Orange can be extended with additional modules. We currently provide an extensive collection of methods for

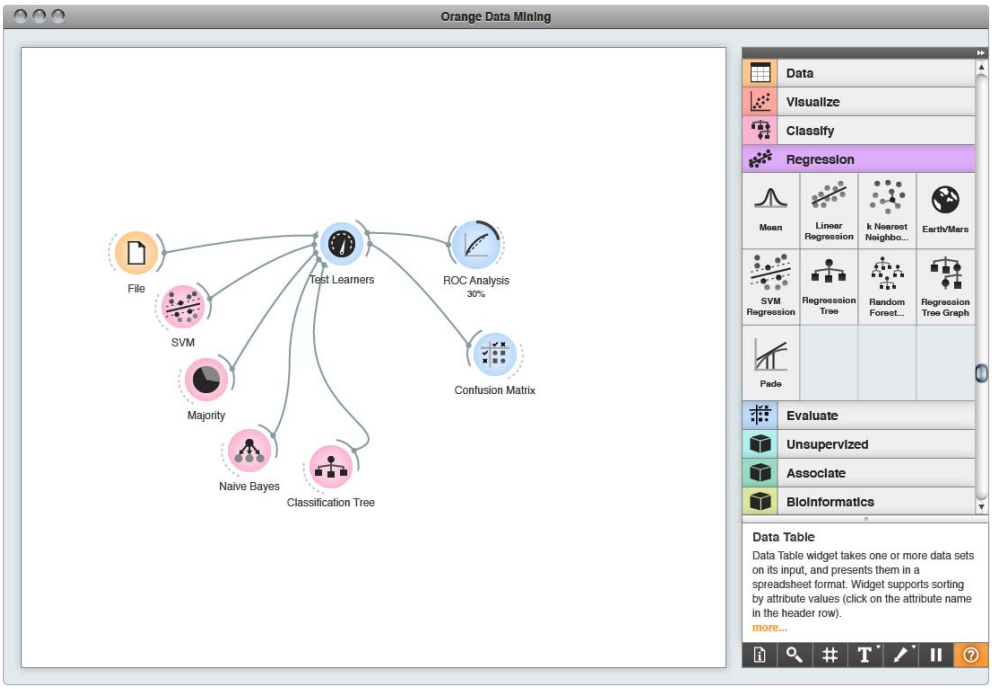

Figure 1. The design of a graphical interface for the upcoming version of Orange (by Peter Čuhalev). The pipeline in the figure reads the data (File widget), sends it to Test Learners which cross validates four different learners on its input and sends it for graphical analysis of an ROC curve and to widget with interactive display of confusion matrix. The data visualizations are not shown in the Figure and are obtained by double-clicking the selected widget.

bioinformatics, as well as modules for text mining and multi-target learning.

The system is being actively developed by a group of about dozen members and collaborators of the Laboratory of bioinformatics, occasionally helped with students of the Faculty of Computer and Information Science and abroad. Development has also been sponsored by Google through its Summer of Code schema.

Orange has been used in science, industry and for teaching. Scientifically, it is used as a testing platform for new machine learning algorithms, as well as for implementing new techniques in genetics and other fields of bioinformatics. At present, the most notable industrial partner is Astra-Zeneca, a pharmaceutical giant, who uses Orange in drug development and sponsors the development of several related parts of Orange. At Jožef Stefan Institute, the visual programming interface has been upgraded in Orange4WS[7] to support service-oriented architectures. Finally, Orange is being used for teaching courses in machine learning and data mining in countries around the world, including the US, Italy, France, Japan, Turkey, Cuba and Peru.

4 FUTURE DEVELOPMENTS

The landscape of Python's libraries has been strongly affected by reorganizing the Numeric and its unfortunate successor numarray into numpy[8]. This has become a standard library for scientific computing in Python. numpy provides arrays of arbitrary dimensions and linear algebra routines from BLAS and ATLAS, and another library, scipy, adds many other common scientific routines, from statistical functions to fast Fourier transforms. A library called scikit-learn is another software that is build on numpy and relies on its fast vectorized operations; scikit-learn[9] contains high quality implementations of most machine learning algorithms and is widely used by the community.

Today we recognize that the power of Orange is not as much in its machine learning algorithms, which should in fact be complemented by several superior ones in scikit-learn, but rather in the way in which these algorithms are packed and exposed to Python scripting in a simpler form. Beyond that, an even stronger feature of Orange is its graphical user interface and visual programming environment, which other Python-based libraries for machine learning lack.

We are intensely working on a new version of Orange in which we will replace the entire C++ core with routines in numpy, scipy, scikit-learn and similar 3rd party open source libraries for Python. This will encourage the contributions from outside of the group, and at the same time allow us to concentrate on development of just those parts in which we are most experienced and in which Orange excels. With it and planned for early 2013 is also a revamped user interface (Figure 1) and http://myflow.io, a platform with a web-based interface to Orange.

[7] http://orange4ws.ijs.si
[8] http://numpy.scipy.org
[9] http://scikit-learn.org

Acknowledgements
Development of Orange is a team effort of many developers and we thank them all for enthusiasm and support. Major contributions to the package were in the past years made by Aleš Erjavec, Gregor Leban, Tomaž Curk, Marko Toplak, Miha Štajdohar, Anže Starič, Matija Polajnar, Lan Žagar, Jure Žbontar, Mitar Milutinović, Lan Umek, Črt Gorup, Martin Možina, Gregor Rot, Aleks Jakulin and designers Peter Čuhalev and Roman Ražman.

References

[1] Quinlan, *C4.5: programs for machine learning* , San Mateo, Calif.: Morgan Kaufmann, 1993.

[2] P. Clark and T. Niblett, "The CN2 induction algorithm", *Machine Learning*, vol. 3, 1987, p. 261-283.

[3] B. Cestnik, I. Kononenko and I. Bratko, "ASSISTANT 86: A knowledge elicitation tool for sophisticated users", *Progress in machine learning*, I. Bratko and N. Lavrac, ed., Sigma Press, 1987.

[4] C. Brunk, J. Kelly and R. Kohavi, "MineSet: an integrated system for data mining", *KDD-97*, 1997, p. 135-138.

[5] T. Curk, J. Demsar, Q. Xu, G. Leban, U. Petrovic, I. Bratko, G. Shaulsky and B. Zupan, "Microarray data mining with visual programming", *Bioinformatics*, vol. 21, Feb. 2005, p. 396-8.