**From Experimental Machine Learning to Interactive Data Mining**

Orange exists thanks to a number of open source projects. Python is used as a scripting language that connects the core components coded in C++. Qt saved us from having to prepare and maintain separate graphical interfaces for MS Windows, Linux and Mac OS X. Python to Qt interface is taken care by PyQt. Additional packets used are Qwt (a set of Qt widgets for technical applications) among with PyQwt that allows us to use it from Python, and Numeric Python (a linear algebra module).

Quite a few years ago, we were (each!) writing our own code for attribute scoring, decision tree induction, ten-fold cross validation and alike (not to mention routines for loading the data and pretty-printing). We got bored. Knowing that coding of the basic set of tools from the ground up was within the job description of just about any researcher in machine learning did not help. At the time quite a few machine learning programs like C4.5 and CN2 were available, but they were coded separately, used different data file formats, and were incompatible in every other respect. There were very few machine learning suites available, which did not offer much in terms of easy prototyping and flexibility in experimenting.

Orange is a library of C++ core objects and routines that includes a large variety of standard and not-so-standard machine learning and data mining algorithms, plus routines for data input and manipulation. Orange is also a scriptable environment for fast prototyping of new algorithms and testing schemes. It is a collection of Python-based modules that sit over the core library and implement some functionality for which execution time is not crucial and which is easier done in Python than in C++. This includes a variety of tasks such as pretty-print of decision trees, attribute subset, bagging and boosting, and alike. Orange is also a set of graphical widgets that use methods from core library and Orange modules and provide a nice user’s interface. Widgets support signal-based communication and can be assembled together into an application by a visual programming tool called Orange Canvas.

All these together make an Orange, a comprehensive, component-based framework for machine learning and data mining. Orange is intended for both experienced users and researchers in machine learning who want to develop and test their own algorithms while reusing as much of the code as possible, and for those just entering the field who can either write short Python scripts for data analysis or enjoy a powerful while easy-to-use visual programming environment.

Orange core objects and Python modules support various data mining tasks that span from data preprocessing to modeling and evaluation. Among other are techniques for:

* Data input, providing the support for various popular data formats
* Data manipulation and preprocessing, like sampling, filtering, scaling, discretization, construction of new attributes, and alike
* Methods for development of classification models, including classification trees, naïve Bayesian classifier, instance-based approaches, logistic regression and support vector machines
* Regression methods, including linear regression, regression trees, and instance-based approaches
* Various wrappers, like those for calibration of probability predictions of classification models
* Ensemble approaches, like boosting and bagging
* Various state-of-the-art constructive induction methods, including function decomposition
* Association rules and data clustering methods
* Evaluation methods, different hold-out schemes and range of scoring methods for prediction models including classification accuracy, AUC, Brier score, and alike. Various hypothesis testing approaches are also supported
* Methods to export predictive models to PMML.