Exposé:
An Ontology for Data Mining Experiments

Joaquin Vanschoren[1] and Larisa Soldatova[2]

[1] Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001 Leuven, Belgium,
joaquin.vanschoren@cs.kuleuven.be
[2] Aberystwyth University, Llandinum Bldg, Penglais, SY23 3DB Aberystwyth, UK,
lss@aber.ac.uk

Abstract. Research in machine learning and data mining can be speeded up tremendously by moving empirical research results out of people's heads and labs, onto the network and into tools that help us structure and filter the information. This paper presents Exposé, an ontology to describe machine learning experiments in a standardized fashion and support a *collaborative* approach to the analysis of learning algorithms. Using a common vocabulary, data mining experiments and details of the used algorithms and datasets can be shared between individual researchers, software agents, and the community at large. It enables open repositories that collect and organize experiments by many researchers. As can been learned from recent developments in other sciences, such a free exchange and reuse of experiments requires a clear representation. We therefore focus on the design of an ontology to express and share experiment meta-data with the world.

1 Introduction

Research in machine learning is inherently empirical. Whether the goal is to develop better learning algorithms or to create appropriate data mining workflows for new sources of data, running the right experiments and correctly interpreting the results is crucial to build up a thorough understanding of learning processes.

Running those experiments tends to be quite laborious. In the case of evaluating a new algorithm, pictured in Figure 1, one needs to search for datasets, preprocessing algorithms, (rival) learning algorithm implementations and scripts for algorithm performance estimation (e.g. cross-validation). Next, one needs to set up a wide range of experiments: datasets need to be preprocessed and algorithm parameters need to be varied, each of which requires much expertise. This easily amounts to a large range of experiments representing days, if not weeks of work, while only averaged results will ever be published. Any other researcher willing to verify the published results or test additional hypothesis will have to start again from scratch, repeating the same experiments instead of simply reusing them.

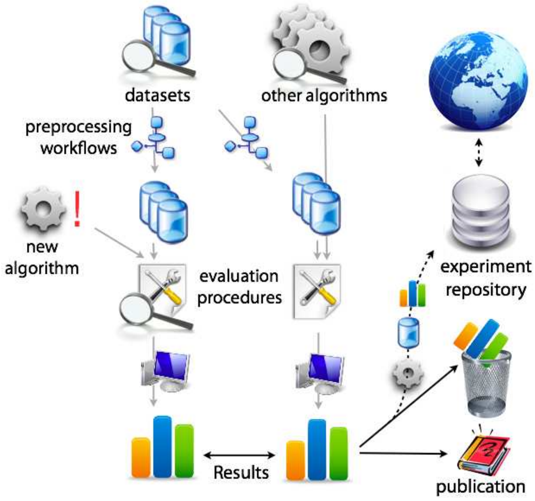

Fig. 1. A typical experimental workflow in machine learning research.

1.1 Generalizability and Interpretability

Moreover, in order to ensure that results are generally valid, the empirical evaluation also needs to cover many different conditions. These include various parameter settings and various kinds of datasets, e.g. differing in size, skewness, noisiness, and various workflows of preprocessing techniques. Unfortunately, because of the amount of work involved in empirical evaluation, many studies will not explore these conditions thoroughly, limiting themselves to algorithm benchmarking. It has long been recognized that such studies are in fact only ‘case studies’ [1], and should be interpreted with caution.

Sometimes, overly general conclusions can be drawn. In time series analysis research, many studies were shown to be biased toward the datasets being used, leading to contradictory results [16]. Moreover, it has been shown that the relative performance of learning algorithms depends heavily on the amount of sampled training data [23, 29], and is also easily dominated by the effect of parameter optimization and feature selection [14].

As such, there are good reasons to thoroughly explore different conditions, or at least to clearly state under which conditions certain conclusions may or may not hold. Otherwise, it is very hard for other researchers to correctly interpret the results, thus possibly creating a false sense of progress [11]:

> *...no method will be universally superior to other methods: relative superiority will depend on the type of data used in the comparisons, the particular data sets, the performance criterion and a host of other factors. [...] an apparent superiority in classification accuracy, obtained in laboratory conditions, may not translate to a superiority in real-world conditions...*

1.2 A collaborative approach

In this paper, we advocate a much more dynamic, *collaborative* approach to experimentation, in which all experiment details can be freely shared in repositories (see the dashed arrow in Fig. 1), linked together with other studies, augmented with measurable properties of algorithms and datasets, and immediately reused by researchers all over the world. Any researcher creating empirical meta-data should thus be able to easily share it with others and in turn reuse any prior results of interest. Indeed, by reusing prior results we can avoid unnecessary repetition and speed up scientific research. This enables large-scale, very generalizable machine learning studies which are prohibitively expensive to start from scratch. Moreover, by bringing the results of many studies together, we can obtain an increasingly detailed picture of learning algorithm behavior. If this meta-data is also properly organized, many questions about machine learning algorithms can be answered on the fly by simply writing a query to a database [29]. This also drastically facilitates meta-learning studies that analyze the stored empirical meta-data to find or useful patterns in algorithm performance [28].

1.3 Ontologies

The use of such public experiment repositories is common practice in many other scientific disciplines. To streamline the sharing of experiment data, they created unambiguous description languages, based on a careful analysis of the concepts used within a domain and their relationships. This is formally represented in *ontologies* [5, 13]: machine manipulable domain models in which each concept (class) is clearly described. They provide an unambiguous vocabulary that can be updated and extended by many researchers, thus harnessing the "collective intelligence" of the scientific community [10]. Moreover, they express scientific concepts and results in a formalized way that allows software agents to interpret them correctly, answer queries and automatically organize all results [25].

In this paper, we propose an ontology designed to adequately record machine learning experiments and workflows in a standardized fashion, so they can be shared, collected and reused. Section 2 first discusses the use of ontologies in other sciences to share experiment details and then covers previously proposed ontologies for data mining. Next, we present Exposé, a novel ontology for machine learning experimentation, in Section 3. Section 4 concludes.

2 Previous work

2.1 e-Sciences

Ontologies have proven very successful in bringing together the results of researchers all over the world. For instance, in astronomy, ontologies are used to build *Virtual Observatories* [7, 27], combining astronomical data from many different telescopes. Moreover, in bio-informatics, the Open Biomedical Ontology

(OBO) Foundry[3] defines a large set of consistent and complementary ontologies for various subfields, such as microarray data[4], and genes and their products [2].

As such, they create an "open scientific culture where as much information as possible is moved out of people's heads and labs, onto the network and into tools that can help us structure and filter the information" [20].

Ironically, while machine learning and data mining have been very successful in speeding up scientific progress in these fields by discovering useful patterns in a myriad of collected experimental results, machine learning experiments themselves are currently not being documented and organized well enough to engender the same automatic discovery of insightful patterns that may speed up the design of new data mining algorithms or workflows.

2.2 Data mining ontologies

Recently, the design of ontologies for data mining attracted quite a bit of attention, resulting in many ontologies for various goals.

OntoDM [22] is a general ontology for data mining with the aim of providing a unified framework for data mining research. It attempts to cover the full width of data mining research, containing high-level classes, such as data mining tasks and algorithms, and more specific classes related to certain subfields, such as constraints for constraint-based data mining.

EXPO [26] is a top-level ontology that models scientific experiments in general, so that empirical research can be uniformly expressed and automated. It covers classes such as hypotheses, (un)controlled variables, experimental designs and experimental equipment.

DAMON (DAta Mining ONtology) [4], is a taxonomy meant to offer domain experts a way to look up tasks, methods and software tools given a certain goal.

KDDONTO [8] is an OWL-DL ontology also built to discover suitable KD algorithms and to express workflows of KD processes. It covers the inputs and outputs of the algorithms and any pre- and postconditions for their use.

KD ontology [31] describes planning-related information about datasets and KD algorithms. It is used in conjunction with an AI planning algorithm: pre- and postconditions of KD operators are converted into standard PDDL planning problems [18]. It is used in an extension of the Orange toolkit to automatically plan KD workflows [32].

The **DMWF** ontology [17] also describes all KD operators with their in- and outputs and pre- and postconditions, and is meant to be used in a KD support system that generates (partial) workflows, checks and repairs workflows built by users, and retrieves and adapts previous workflows.

DMOP, the Data Mining Ontology for Workflow Optimization [12], models the internal structure of learning algorithms, and is explicitly designed to support algorithm selection. It covers classes such as the structure and parameters of predictive models, the involved cost functions and optimization strategies.

[3] http://www.obofoundry.org/

[4] http://www.mged.org/ontology

3 The Exposé ontology

In this section, we describe Exposé, an ontology for machine learning experimentation. It is meant to be used in conjunction with *experiment databases* (ExpDBs) [3, 29, 28]: databases designed to collect the details of these experiments, and to intelligently organize them in online repositories to enable fast and thorough analysis of a myriad of collected results. In this context, Exposé supports the accurate recording and exchange of data mining experiments and workflows. It has been 'translated' into an XML-based language, called ExpML, to describe experiment workflows and results in detail [30]. Moreover, it clearly defines the semantics of data mining experiments stored in the experiment database, so that a very wide range of questions on data mining algorithm performance can be answered through querying [29]. Many examples can be found in previous papers [29, 30]. Finally, although we currently use a relational database, Exposé will clearly be instrumental in RDF databases, allowing even more powerful queries. It thus supports reasoning with the data, meta learning, data integration, and also enables logical consistency checks.

For now, Exposé focuses on supervised classification on propositional datasets. It is also important to note that, while it has been influenced and adapted by many researchers, it is a straw-man proposal that is intended to instigate discussion and attract wider involvement from the data mining community. It is described in the OWL-DL ontology language [13], and can be downloaded from the experiment database website (`http://expdb.cs.kuleuven.be`).

We first describe the design guidelines used to develop Exposé, then its top-level classes, and finally the parts covering experiments, experiment contexts, evaluation metrics, performance estimation techniques, datasets, and algorithms.

3.1 Ontology design

In designing Exposé, we followed existing guidelines for ontology design [21, 15]:

Top-level ontologies It is considered good practice to start from generally accepted classes and relationships (properties) [22]. We started from the Basic Formal Ontology (BFO)[5] covering top-level scientific classes and the OBO Relational Ontology (RO)[6] offering a predefined set of properties.

Ontology reuse If possible, other ontologies should be reused to build on prior knowledge and consensus. We directly reuse several general machine learning related classes from OntoDM [22], experimentation-related classes from EXPO [26], and classes related to internal algorithm mechanisms from DMOP [12]. We wish to integrate Exposé with existing ontologies, so that it will evolve with them as they are extended further.

Design patterns Ontology design patterns[7] are reusable, successful solutions to recurrent modeling problems. For instance, a learning algorithm can sometimes act as a base-learner for an ensemble learner. This is a case of an

[5] http://www.ifomis.org/bfo

[6] http://www.obofoundry.org/ro/

[7] http://ontologydesignpatterns.org

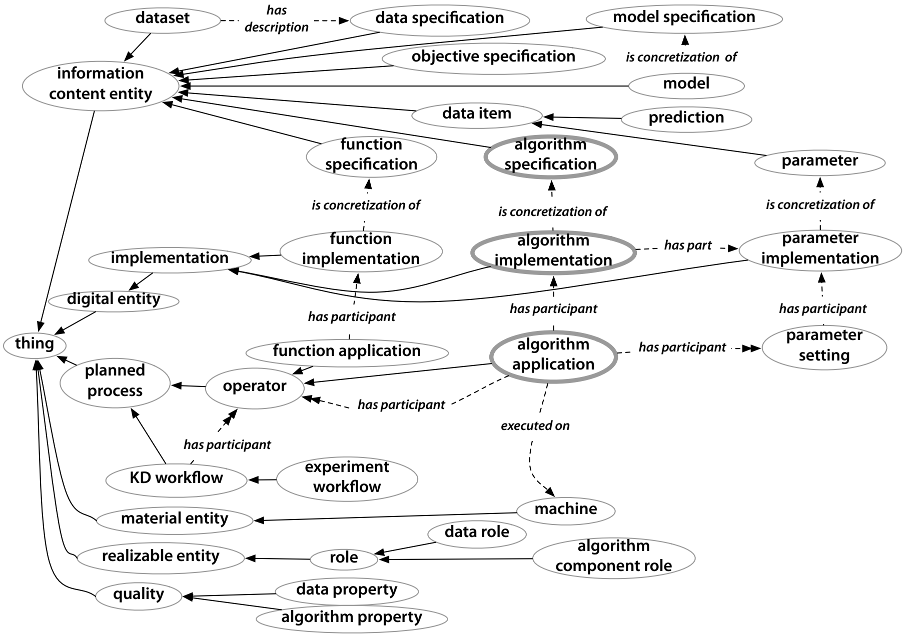

Fig. 2. An overview of the top-level classes in the Exposé ontology.

agent-role pattern, and a predefined property, 'realizes', is used to indicate which entities are able to fulfill a certain role.

Quality criteria General criteria include *clarity*, *consistency*, *extensibility* and *minimal commitment*. These criteria are rather qualitative, and were only evaluated through discussions with other researchers.

3.2 Top-level View

Figure 2 shows the most important top-level classes and properties, many of which are inherited from the OntoDM ontology [22], which in turn reuses classes from OBI[8] (i.e., *planned process*) and IAO[9] (i.e. *information content entity*). The full arrows symbolize an 'is-a' property, meaning that the first class is a subclass of the second, and the dashed arrows symbolize other common properties. Double arrows indicate one-to-many properties, for instance, an *algorithm application* can have many *parameter settings*.

The three most important categories of classes are *information content entity*, which covers datasets, models and abstract specifications of objects (e.g. algorithms), *implementation*, and *planned process*, a sequence of actions meant to achieve a certain goal. When describing experiments, this distinction is very important. For instance, the class 'C4.5' can mean the abstract algorithm, a specific implementation or an execution of that algorithm with specific parameter settings, and we want to distinguish between all three.

[8] http://obi-ontology.org

[9] http://code.google.com/p/information-artifact-ontology

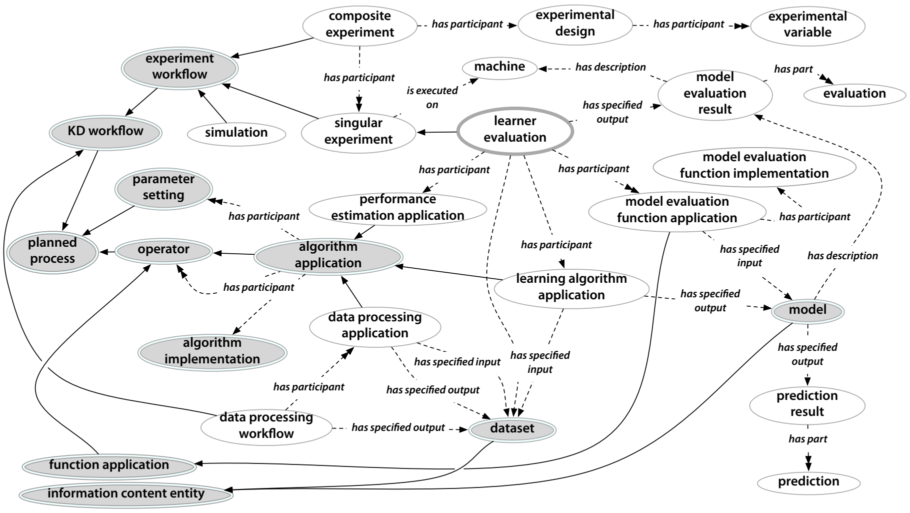

Fig. 3. Experiments in the Exposé ontology.

As such, ambiguous classes such as 'learning algorithm' are broken up according to different interpretations (indicated by bold ellipses in Fig. 2): an abstract *algorithm specification* (e.g. in pseudo-code), a concrete *algorithm implementation*, code in a certain programming language with a version number, and a specific *algorithm application*, a deterministic function with fixed parameter settings, run on a specific *machine* with an actual input (a *dataset*) and output (a *model*), also see Fig. 3. The same distinction is used for other algorithms (for data preprocessing, evaluation or model refinement), mathematical functions (e.g. the kernel used in an SVM), and parameters, which can have different names in different implementations and different value settings in different applications. Algorithm and function applications are *operators* in a KD workflow, and can even be participants of another algorithm application (e.g., a kernel or a base-learner), i.e. they can be part of the inner workflow of an algorithm.

Finally, there are also *qualities*, properties of a specific dataset or algorithm (see Figs. 6 and 7), and *roles* indicating that an element assumes a (temporary) role in another process: an algorithm can act as a base-learner in an ensemble, a function can act as a distance function in a learning algorithm, and a dataset can be a training set in one experiment and a test set in the next.

3.3 Experiments

Figure 3 shows the ontological description of experiments, with the top-level classes from Fig. 2 drawn in filled double ellipses. Experiments are defined as *workflows*, which allows the description of many kinds of experiments. Some (composite) experiments can also consist of many smaller (singular) experiments, and can use a particular *experiment design* [19] to investigate the effects of various *experimental variables*, e.g. parameter settings.

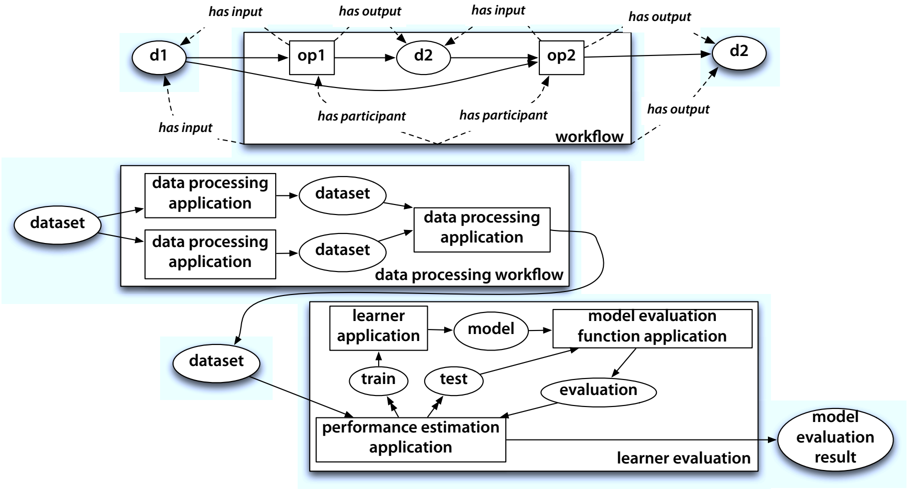

Fig. 4. Workflow structure and an example experiment workflow.

We will now focus on a particular kind of experiment: a *learner evaluation* (indicated by a bold ellipse). This type of experiment applies a specific learning algorithm (with fixed parameters) on a specific input dataset and evaluates the produced model by applying one or several model evaluation functions, e.g. predictive accuracy. In predictive tasks, a performance estimation technique, e.g. 10-fold cross-validation, is applied to generate training- and test sets, evaluate the resulting models and aggregate the results. After it is executed on a specific *machine*, it will output a *model evaluation result* containing the outcomes of all evaluations and, in the case of predictive algorithms, the (probabilistic) predictions made by the models. Models are also generated by applying the learning algorithm on the entire dataset.

Finally, more often than not, the dataset will have to be preprocessed first. Again, by using workflows, we can define how various *data processing applications* preprocess the data before it is passed on to the learning algorithm. Figure 4 illustrates such a workflow. The top of the figure shows that it consists of participants (operators), which in turn have inputs and outputs (shown in ovals): datasets, models and model evaluation results. Workflows themselves also have inputs and outputs, and we can define specialized sub-workflows. A *data processing workflow* is a sequence of data processing steps. The center of Fig. 4 shows one with three preprocessors. A *learner evaluation* workflow takes a dataset as input and applies performance estimation techniques (e.g. 10-fold cross-validation) and model evaluation functions (e.g. the area under the ROC curve) to evaluate a specific learner application. Of course, there are other types of learner evaluations, both finer ones, e.g. a singular train-test experiment, and more complex ones, e.g. doing an internal model selection to find the optimal parameter settings.

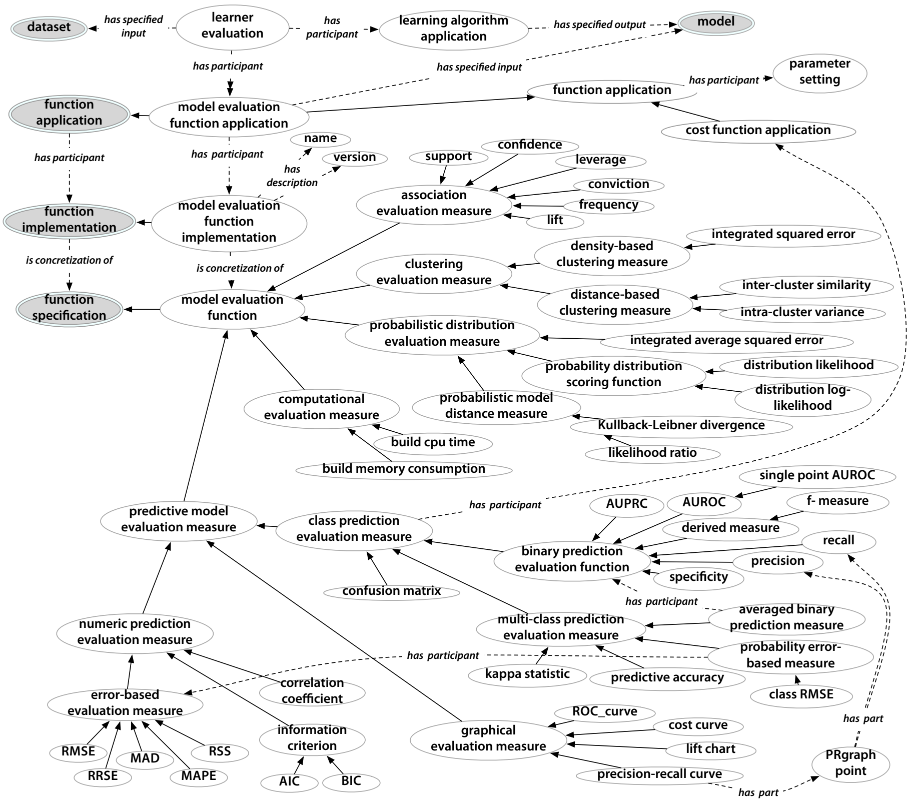

Fig. 5. Learner evaluation measures in the Exposé ontology.

3.4 Experiment context

Although outside the scope of this paper, Exposé also models the *context* in which scientific investigations are conducted. Many of these classes are originally defined in the EXPO ontology [26]. They include authors, references to publications and the *goal*, *hypotheses* and *conclusions* of a study. It also defines (un)controlled or (in)dependent *experimental variables*, and various *experimental designs* [19] defining which values to assign to each of these variables.

3.5 Learner evaluation

To describe algorithm evaluations, Exposé currently covers 96 performance measures used in various learning tasks, some of which are shown in Fig. 5. In some tasks, all available data is used to build a model, and properties of that model are measured to evaluate it, e.g., the *inter-cluster similarity* in clustering. In binary classification, the predictions of the models are used, e.g., *predictive accuracy*, *precision* and *recall*. In multi-class problems, the same measures can be used by transforming the multi-class prediction into c binary predictions, and averaging the results over all classes, weighted by the number of examples in each

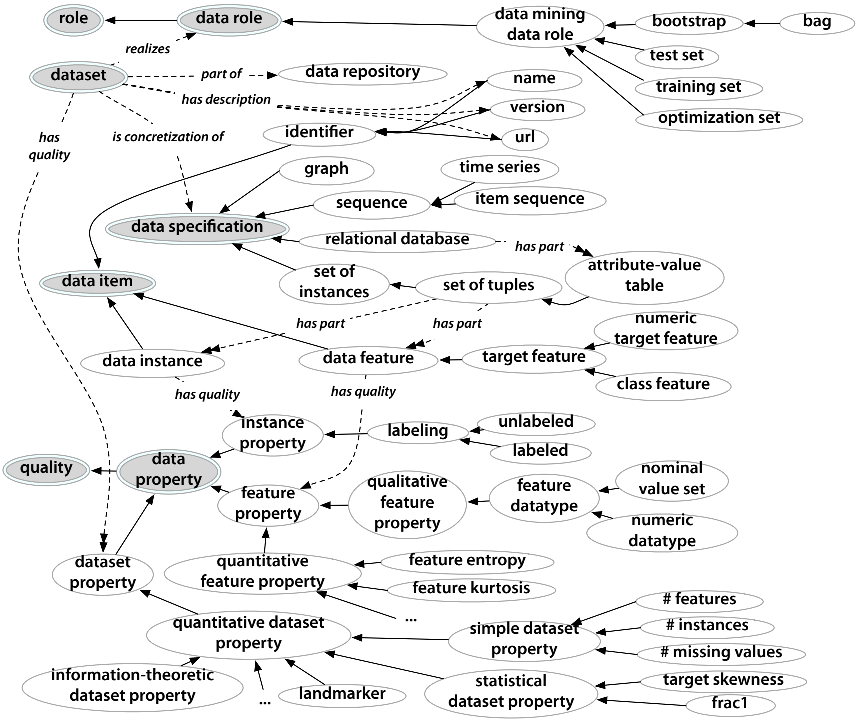

Fig. 6. Datasets in the Exposé ontology.

class. Regression measures, e.g., *root mean squared error* (RMSE) can also be used in classification by taking the difference between the actual and predicted class probabilities. Finally, *graphical evaluation measures*, such as *precision-recall curves*, *ROC-curves* or *cost-curves*, provide a much more detailed evaluation. Many definitions of these metrics exist, so it is important to define them clearly.

Although not shown here, Exposé also covers several *performance estimation algorithms*, such as *k-fold* or *5x2 cross-validation*, and statistical significance tests, such as the paired t-test (by resampling, 10-fold cross-validation or 5x2 cross-validation) [9] or tests on multiple datasets [6].

3.6 Datasets

Figure 6 shows the most important classes used to describe datasets.

Specification. The *data specification* (in the top part of Fig. 6) describes the structure of a dataset. Some subclasses are graphs, sequences and sets of instances. The latter can have instances of various types, e.g., tuples, in which case it can have a number of *data features* and *data instances*. For other types of data this specification will have to be extended. Finally, a dataset has descriptions, such as name, version and download url to make it easily retrievable.

Roles. A specific dataset can play different *roles* in different experiments (top of Fig. 6). For instance, it can be a training set in one evaluation and a test set in the next.

Data properties. As said before, we wish to link all empirical results to theoretical metadata, called *properties*, about the underlying datasets to perform meta-learning studies. These *data properties* are shown in the bottom half of Fig. 6, and may concern individual instances, individual features or the entire dataset. We define both *feature properties* such as feature skewness or mutual information with the target feature, as well as general *dataset properties* such as the number of attributes and landmarkers [24].

3.7 Algorithms

Algorithms can perform very differently under different configurations and parameter settings, so we need a detailed vocabulary to describe them. Figure 7 shows how algorithms and their configurations are expressed in our ontology. From top to bottom, it shows a taxonomy of different types of algorithms, the different internal operators they use (e.g. kernel functions), the definition of algorithm implementations and applications (see Sect. 3.2) and algorithm properties (only two are shown).

Algorithm implementations. Algorithm implementations are described with all information needed to retrieve and use them, such as their name, version, url, and the library they belong to (if any). Moreover, they have implementations of algorithm parameters and can have *qualities*, e.g. their susceptibility to noise.

Algorithm composition. Some algorithms use other algorithms or mathematical functions, which can often be selected (or plugged in) by the user. These include base-learners in ensemble learners, distance functions in clustering and nearest neighbor algorithms and kernels in kernel-based learning algorithms. Some algorithm implementations also use internal data processing algorithms, e.g. to remove missing values. In Exposé, any operator can be a participant of an algorithm application, combined in internal workflows with in- and outputs. Depending on the algorithm, operators can fulfill (*realize*) certain predefined *roles* (center of Fig. 7).

Algorithm mechanisms. Finally, to understand the performance differences between different types of algorithms, we need to look at the internal learning mechanisms on which they are built. These include the kind of models that are built (e.g. decision trees), how these models are optimized (e.g. the heuristic used, such as information gain) and the decision boundaries that are generated (e.g. axis-parallel, piecewise linear ones in the case of non-oblique decision trees). These classes, which extend the algorithm definitions through specific properties (e.g. *has model structure*), are defined in the DMOP ontology [12], so they won't be repeated here.

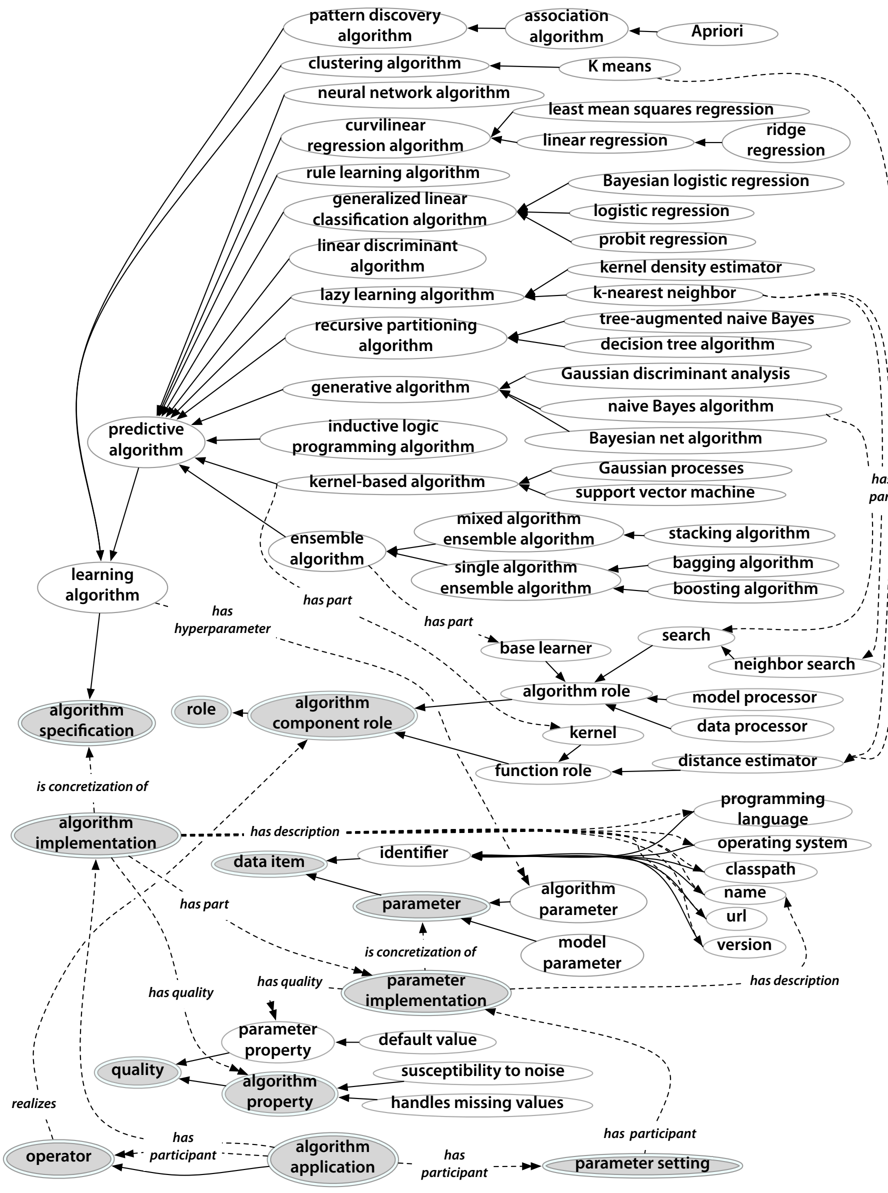

Fig. 7. Algorithms and their configurations in the Exposé ontology.

4 Conclusions

We have presented Exposé, an ontology for data mining experiments. It is complementary to other data mining ontologies such as OntoDM [22], EXPO [26], and DMOP [12], and covers data mining experiments in fine detail, including the experiment context, evaluation metrics, performance estimation techniques, datasets, and algorithms. It is used in conjunction with *experiment databases* (ExpDBs) [3, 29, 28], to engender a *collaborative* approach to empirical data mining research, in which experiment details can be freely shared in repositories, linked together with other studies, and immediately reused by researchers all over the world. Many illustrations of the uses of Exposé to share, collect and query for experimental meta-data can be found in prior work [3, 29, 30].

References

1. Aha, D.: Generalizing from case studies: A case study. Proceedings of the Ninth International Conference on Machine Learning pp. 1–10 (1992)
2. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G.: Gene ontology: tool for the unification of biology. nature genetics 25, 25–29 (2000)
3. Blockeel, H., Vanschoren, J.: Experiment databases: Towards an improved experimental methodology in machine learning. Lecture Notes in Computer Science 4702, 6–17 (2007)
4. Cannataro, M., Comito, C.: A data mining ontology for grid programming. First International Workshop on Semantics in Peer-to-Peer and Grid Computing at WWW 2003 pp. 113–134 (2003)
5. Chandrasekaran, B., Josephson, J.: What are ontologies, and why do we need them? IEEE Intelligent systems 14(1), 20–26 (1999)
6. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)
7. Derriere, S., Preite-Martinez, A., Richard, A.: UCDs and ontologies. ASP Conference Series 351, 449 (2006)
8. Diamantini, C., Potena, D., Storti, E.: Kddonto: An ontology for discovery and composition of kdd algorithms. Proceedings of the 3rd Generation Data Mining Workshop at the 2009 European Conference on Machine Learning (2009)
9. Dietterich, T.: Approximate statistical tests for comparing supervised classification learning algorithms. Neural computation 10(7), 1895–1923 (1998)
10. Goble, C., Corcho, O., Alper, P., Roure, D.D.: e-science and the semantic web: A symbiotic relationship. Lecture Notes in Computer Science 4265, 1–12 (2006)
11. Hand, D.: Classifier technology and the illusion of progress. Statistical Science 21(1), 114 (2006)
12. Hilario, M., Kalousis, A., Nguyen, P., Woznica, A.: A data mining ontology for algorithm selection and meta-mining. Proceedings of the ECML/PKDD09 Workshop on 3rd generation Data Mining (SoKD-09) pp. 76–87 (2009)
13. Horridge, M., Knublauch, H., Rector, A., Stevens, R., Wroe, C.: A practical guide to building OWL ontologies using Protege 4 and CO-ODE tools. The University of Manchester (2009)

14. Hoste, V., Daelemans, W.: Comparing learning approaches to coreference resolution. there is more to it than bias. Proceedings of the Workshop on Meta-Learning (ICML-2005) pp. 20–27 (2005)
15. Karapiperis, S., Apostolou, D.: Consensus building in collaborative ontology engineering processes. Journal of Universal Knowledge Management 1(3), 199–216 (2006)
16. Keogh, E., Kasetty, S.: On the need for time series data mining benchmarks: A survey and empirical demonstration. Data Mining and Knowledge Discovery 7(4), 349–371 (2003)
17. Kietz, J., Serban, F., Bernstein, A., Fischer, S.: Towards cooperative planning of data mining workflows. Proceedings of the Third Generation Data Mining Workshop at the 2009 European Conference on Machine Learning (ECML 2009) pp. 1–12 (2009)
18. Klusch, M., Gerber, A., Schmidt, M.: Semantic web service composition planning with owls-xplan. Proceedings of the First International AAAI Fall Symposium on Agents and the Semantic Web (2005)
19. Kuehl, R.: Design of experiments: statistical principles of research design and analysis. Duxbury Press (1999)
20. Nielsen, M.: The future of science: Building a better collective memory. APS Physics 17(10) (2008)
21. Noy, N., McGuinness, D.: Ontology development 101: A guide to creating your first ontology. Stanford University (2002)
22. Panov, P., Soldatova, L., Dzeroski, S.: Towards an ontology of data mining investigations. Lecture Notes in Artificial Intelligence 5808, 257–271 (2009)
23. Perlich, C., Provost, F., Simonoff, J.: Tree induction vs. logistic regression: A learning-curve analysis. Journal of Machine Learning Research 4, 211–255 (2003)
24. Pfahringer, B., Bensusan, H., Giraud-Carrier, C.: Meta-learning by landmarking various learning algorithms. Proceedings of the Seventeenth International Conference on Machine Learning pp. 743–750 (2000)
25. Sirin, E., Parsia, B.: SPARQL-DL: SPARQL query for OWL-DL. Third International Workshop on OWL Experiences and Directions (OWLED 2007) (2007)
26. Soldatova, L., King, R.: An ontology of scientific experiments. Journal of the Royal Society Interface 3(11), 795–803 (2006)
27. Szalay, A., Gray, J.: The world-wide telescope. Science 293, 2037–2040 (2001)
28. Vanschoren, J., Blockeel, H., Pfahringer, B., Holmes, G.: Experiment databases: Creating a new platform for meta-learning research. Proceedings of the ICML/UAI/COLT Joint Planning to Learn Workshop (PlanLearn08) pp. 10–15 (2008)
29. Vanschoren, J., Pfahringer, B., Holmes, G.: Learning from the past with experiment databases. Lecture Notes in Artificial Intelligence 5351, 485–492 (2008)
30. Vanschoren, J., Soldatova, L.: Collaborative meta-learning. Proceedings of the Third Planning to Learn Workshop at the 19th European Conference on Artificial Intelligence (2010)
31. Záková, M., Kremen, P., Zelezný, F., Lavrac, N.: Planning to learn with a knowledge discovery ontology. Second planning to learn workshop at the joint ICML/COLT/UAI Conference pp. 29–34 (2008)
32. Záková, M., Podpecan, V., Zelezný, F., Lavrac, N.: Advancing data mining workflow construction: A framework and cases using the Orange toolkit. Proceedings of the SoKD-09 International Workshop on Third Generation Data Mining at ECML PKDD 2009 pp. 39–51 (2009)