Agnostic Learning *vs.* Prior Knowledge Challenge

Isabelle Guyon, Amir Saffari, and Gideon Dror

***Abstract*— "When everything fails, ask for additional domain knowledge" is the current motto of machine learning. Therefore, assessing the real added value of prior/domain knowledge is a both deep and practical question. Most commercial data mining programs accept data pre-formatted as a table, each example being encoded as a fixed set of features. Is it worth spending time engineering elaborate features incorporating domain knowledge and/or designing ad hoc algorithms? Or else, can off-the-shelf programs working on simple features encoding the raw data without much domain knowledge do as well or better than skilled data analysts? To answer these questions, we organized a challenge for IJCNN 2007. The participants were allowed to compete in two tracks: The "prior knowledge" track, for which they had access to the original raw data representation and as much knowledge as possible about the data, and the "agnostic learning" track for which they were forced to use data pre-formatted as a table with dummy features. The challenge web site remains open for post-challenge submissions: http://www.agnostic.inf.ethz.ch/.**

I. Introduction

It has been a long time philosophical and scientific debate whether or not the children brains are a "tabula rasa" virgin of prior knowledge about their environment. While it is still unclear whether this hypothesis holds or not for the neocortex, it cannot be refuted that specialized cortexes connected to sensory inputs have evolved over millions of years to process information in a specialized manner [1]. Hence, to perform some of its learning tasks the brain benefits from advanced feature extraction embodying in some sense "prior knowledge". On one hand, such specialized preprocessing allows humans and animals to excel in certain tasks such as face recognition. On the other hand, the brain is also capable of learning without the benefit of such specialized preprocessing. For instance, a human expert can learn to manage an investment portfolio, a task relying upon data representations not necessarily readily available from the sensory cortexes. Perhaps evolution made us gain a preference for "simple solutions" over different problems in a domain to allow us to complete such tasks without much prior knowledge. Learning machines are tools meant to help engineers solve tasks at the expense of as little human labor as possible. Incorporating "prior knowledge" or "domain knowledge" in a learning machine can be fairly labor and expert intensive, so researchers keep trying to improve their predictive models so they provide good performances at the expense of as little human intervention as possible. In the past decades, this has been made possible even in cases where the number of examples is small compared to the dimension of the feature space, with the new generation of regularized learning machines (see *e.g.* [2]). For the purpose of this challenge, we define "prior knowledge" as any form of knowledge about a given task that may be incorporated in the design of a predictive learning system, prior to training from examples. This may include: feature information (type of features, topological relationships between features, indications of feature relevance) and more general information about the nature and goal of the task that can inform about clusters in data, missing data, etc. For example, in a vision task in which images are encoded as gray level pixels, the knowledge of the nature of the features and their topological relationships allows the designer to perform specialized image filtering [3] or use specialized machine learning architectures, such as a convolutional neural network [4]. An alternative used in handwriting recognition is to model the dynamics of handwriting to extract relevant features from on-line data [5]. Another example is the study of DNA or protein sequences. Knowledge about the primary, secondary, and tertiary structure of a molecule improves the identification of active sites. This may be exploited in specialized kernels used with kernel machines [6]. The goal of this challenge is to determine how much can be gained in performance with the availability of prior knowledge, using tasks for which we have already baseline results from previous challenges in an "agnostic learning" setting. Simultaneously, we will try to push the state-of-the-art in the design of the "perfect black box" by letting the participants make entries with agnostic models in a more constrained framework.

Challenges have been lately greatly stimulating for research in machine learning, pattern recognition, and robotics. Robotics contests seem to be particularly popular, with hundreds of events every year [7], the two most visible ones probably being the DARPA grand challenge of autonomous ground vehicle navigation [8] and RoboCup [9] featuring robots involved in various tasks, including playing soccer and rescuing people. In data mining and machine learning, several conferences have been regularly organizing challenges in the past 10 years, including the well established TREC [10] and the KDD cup [11]. More specialized pattern recognition and bioinformatics conference have also their contests (e.g. CASP for protein structure prediction [12] and ICDAR for document analysis [13]. The European network of excellence PASCAL has been sponsoring a number of challenges around hot themes in machine learning [14], which have been punctuating workshop at NIPS and other conferences.

Attracting hundreds of participants and the attention of a public of specialists as well as sometimes the general public, these events have been important is several respects:

- to push the state-of-the art,

Isabelle Guyon is an independent consultant, Berkeley, California, USA.
Amir Saffari is with the ICG, Graz University of Technology, Austria.
Gideon Dror is with the Academic College of Tel-Aviv-Yaffo, Israel.

- to identify techniques which really work,
- to attract new researchers,
- to raise the standards of research,
- to give the opportunity to non-established researchers to make themselves rapidly known.

We are committed to bringing to IJCNN interesting competitions. Last year's challenge for WCCI 2006/IJCNN 2006 attracted 145 participants. This year two challenges are run in parallel: the classification competition described in this paper and a time series prediction challenge [15]. The participation is not quite as high as last year, which we attribute to several factors, including the simultaneous million dollar challenge organized by Netflix [16], which attracted thousands of participants. As of end of January 31^{st}, 30 entrants participated in our challenge, making over 700 development entries and 100 full challenge entries (including results on all five tasks).

Our challenge design is inspired by previous competitions we organized [17], [18]. This year, we are using the same five large data sets as in last year's challenge, but formatted differently (See Section II). The tasks are five classification problems in marketing, handwriting recognition, drug discovery, text processing and ecology. The competition has two parallel tracks: We supply for the "agnostic learning track" data preprocessed in a simple feature-based representation, suitable to use any off-the-shelf machine learning or data mining algorithm. The same preprocessing as last year is used (but a new data split is provided). The participants have no knowledge of the identity of the features in that track. New in this year's competition are the raw data representations supplied for the "prior knowledge track" , which are not necessarily in the form of data tables. For instance, in the drug discovery problem the raw data consists in the three dimensional layout of the drug molecules; in the text processing problem, the raw data are emails. The participants have full knowledge of the meaning of the representation in that track. Therefore, "Prior knowledge" competitors have the opportunity of using domain knowledge to build good predictors and beat last year's "agnostic" results or new "agnostic" entries.

II. Tasks of the challenge

We used five classification tasks spanning a variety of domains and difficulties. We chose data sets that had sufficiently many examples to create a large enough test set to obtain statistically significant results. The input variables are continuous or binary, sparse or dense. Some raw data representations are non feature based. In some problems, the class proportions are very imbalanced The data characteristics are summarized in Table I. A detailed report on the data preparation is available [19].

Briefly, we present the five tasks of the challenge:

1) **ADA, marketing**: The task of ADA is to discover high revenue people from census data. This is a two-class classification problem. The raw data from the census bureau is known as the Adult database in the UCI machine-learning repository [20]. The 14 original attributes (features) include age, workclass, education, education, marital status, occupation, native country, etc. It contains continuous, binary and categorical features. The "prior knowledge track" has access to the original features and their identity. The "agnostic learning track" has access to a preprocessed numeric representation eliminating categorical variables, but the identity of the features is not revealed.
2) **GINA, handwriting recognition (HWR)**: The task of GINA is handwritten digit recognition. The raw data is known as the MNIST dataset [21]. For the "agnostic learning track" we chose the problem of separating two-digit odd numbers from two-digit even numbers. Only the unit digit is informative for that task, therefore at least 1/2 of the features are distracters. Additionally, the pixels that are almost always blank were removed and the pixel order was randomized to hide the feature identity. For the "prior knowledge track", only the informative digit is provided in the original pixel map representation. This is a two class classification problem with sparse continuous input variables, in which each class is composed of several clusters. It is a problems with heterogeneous classes.
3) **HIVA, drug discovery**: The task of HIVA is to predict which compounds are active against the AIDS HIV infection. The original data from the NCI [22] has 3 classes (active, moderately active, and inactive). We brought it back to a two-class classification problem (active vs. inactive), but we provide the original labels for the "prior knowledge track". The compounds are represented by their 3d molecular structure for the prior knowledge track. For the "agnostic track" we represented the data as 2000 sparse binary input variables. The variables represent properties of the molecule inferred from its structure. The problem is therefore to relate structure to activity (a QSAR=quantitative structure-activity relationship problem) to screen new compounds before actually testing them (a HTS=high-throughput screening problem.)
4) **NOVA, text classification**: The data of NOVA come from the 20-Newsgroup dataset [23]. Each text to

TABLE I
DATASETS OF THE CHALLENGE

Dataset	Domain	Num. examples (tr/va/te)	Pos (%)	Num. features Raw	Prep.
ADA	Marketing	4147/415/41471	28.4	14	48
GINA	HWR	3153/315/31532	49.2	784	970
HIVA	Drug disc.	3845/384/38449	3.5	Mol.	1617
NOVA	Text clas.	1754/175/17537	28.5	Text	16969
SYLVA	Ecology	13086/1309/130857	6.2	108	216

The third column shows the number of training, validation, and test examples. The fourth column indicates the percentage of examples of the positive class. The two last columns show the number of features in the raw data representation (for the "prior knowledge track") and the preprocessed data (for the "agnostic learning track"). Non-feature based representations are supplied for HIVA (molecular structure) and NOVA (emails) in the "prior knowledge track".

classify is an email that was posted to one or several newsgroups. We selected the separation of politics and religion topics from all the other topics. The raw data comes as text files for the "prior knowledge track". The topics are provided with the training data in that track. The preprocessed data for the "agnostic learning track" is a sparse binary representation using a bag-of-word with a vocabulary of approximately 17000 words.

5) **SYLVA, ecology**: The task of SYLVA is to classify forest cover types. The forest cover type for 30 x 30 meter cells was obtained from US Forest Service (USFS) Region 2 Resource Information System (RIS) data [24]. We brought it back to a two-class classification problem (classifying Ponderosa pine *vs.* everything else). The "agnostic learning track" data consists in 216 input variables. Each pattern is composed of 4 records: 2 true records matching the target and 2 records picked at random. Thus 1/2 of the features are distracters. The "prior knowledge track" data is identical to the "agnostic learning track" data, except that the distracters are removed and the identity of the features is revealed. For that track, the forest cover original id's are revealed for training data.

In Figure 1, we show the distribution of performance on the test set of the entries who qualified for the final ranking in the "performance prediction challenge" organized last year on the same datasets. We see that the datasets vary in difficulty. HIVA (drug discovery) seems to be the most difficult dataset: the average BER and the spread are high. ADA (marketing) is the second hardest. The distribution is very skewed and has a heavy tail, indicating that a small group of methods "solved" the problem, which was not obvious to others. NOVA (text classification) and GINA (digit recognition) come next. Both datasets have classes containing multiple clusters. Hence, the problems are highly non-linear. This may property of the data explain the very long distribution tails. Finally, SYLVA (ecology) is the easiest dataset, due to the large amount of training data. These histograms will be updated with this year's results when they become available.

III. Rules of the Challenge

We summarize briefly the rules of the challenge. More details are found the challenge FAQ [25].

- **Duration:** The challenge started October 1^{st}, 2006 will end March 1^{st}, 2007 (duration: 21 weeks). At mid-point in the challenge on December 1^{st} a first ranking of the participants for the model selection game was made (see Section IV), but the scores on the test set were not revealed. After the challenge is over, the web site will remain open for post-challenge submissions.
- **Scope:** To enter the final ranking, the competitors have to return results on all five tasks of the challenge, to alleviate the problem of "over-modeling". To facilitate entering results on all tasks, the tasks are restricted to 2-class classification problems.

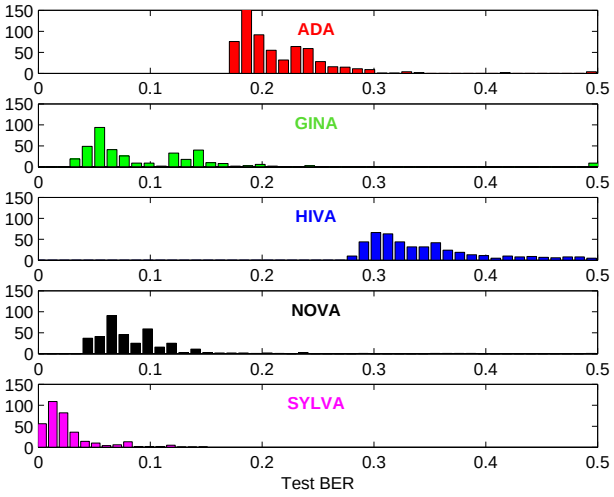

Fig. 1. Distribution of performance on the test set in the IJCNN06 challenge (same datasets). Performance is measured in Balanced Error Rate (BER).

- **Tracks:** The challenge has two tracks:
  1) The "prior knowledge track": The participants to that track are allowed to work from the raw data and use available knowledge about the data.
  2) The "agnostic learning track": The participants to that track are constrained to use a representation with "dummy features" without knowing what these features mean.

  The training/test split is the same in the two tracks, but the example ordering is different in each data subset to hinder matching patterns in the two representations and/or submitting results with the representation prescribed for the other track.
- **On-line feedback:** During a development period, the participants are able to submit results on-line on a small subset of the data (validation set) to get immediate feedback. The final ranking will be performed with separate test data, and the results on test data will be disclosed only after the challenge is over.
- **Baseline results and toolkit:** The organizers provided initial submissions to bootstrap the challenge. A toolkit was provided, including some of the methods performing best in previous challenges (see Appendix).
- **Requisites:** Final submissions eligible for entering the final ranking must include results on all the tasks of the challenge in either track, on the test data. Anonymity may be retained only during the development period. A properly filled out fact sheet will have to accompany the final results. Optionally, if the provided toolkit is used, the submissions will include the predictive models, saved in a designated format (see Section IV).
- **Scoring:** The entries will be ranked for each task and in each track with their balanced error rate (BER) on the test set. The BER is the average of the error rate of the positive class and the error rate of the negative class. The Area Under the *ROC* Curve (AUC) was

also computed, but not used for scoring. The overall ranking in each track will be obtained by averaging the entry ranks on individual tasks. The number of submissions per participant is unlimited, but only the five last "complete" submissions for each participant will be included in the final ranking.

- **Rewards:** The participants are rewarded by publishing their results at IJCNN. In addition, there will be seven $150 prizes. On prize in the "agnostic learning track" for the best overall entry and one prize for each dataset in the "prior knowledge track". The challenge will reward not solely the participants with best quantitative results, but also on the quality of the methodology, including quality of the experiments, originality, and reproducibility, by offering a best paper award.

IV. A MODEL SELECTION GAME

To trigger interest into the challenge, we provided initial baseline results with classical methods. Those methods were implemented with the Spider package developed at the Max Planck Institute for Biological Cybernetics [26]. Our package of methods implemented for the challenge called CLOP (Challenge Learning Object Package) is briefly described in Appendix. The package is available for downloading from the web site of the challenge, but the participants were not forced to use it.

In previous challenges, we noted that different teams using similar classification methods (even sometimes the same software package) obtained very different results. We conjectured that this variance may be due to differences in model selection strategies. To stimulate research in model selection and verify our conjecture, we organized a model selection game, within the present competition. The game consisted in returning results before December 1^{st} 2006 in the "agnostic learning track", using exclusively a set of provided CLOP modules (for a list, see Appendix).

Using CLOP, the game participants closely matched the performances of the best entrants who used their own methods and considerably outperformed the baseline performances provided by the organizers using CLOP models. This proves the efficiency of their model selection techniques.

All model selection methods rely on two basic elements: (1) a scoring function to evaluate the models, and (2) a search algorithm to explore the space of all possible models. In the two last challenges we organized [17], [18], the most successful scoring functions were based on cross-validation; participants relying on the training set error (eventually corrected by some complexity penalty terms) or on the validation set error overfitted the training data. Conversely, cross-validation users could afford searching the model space quite intensively without apparently incurring overfitting problems. Hence, the winners singled themselves out by effectively searching model space. This may be achieved either by brute force grid search using a computer cluster, or by some more refined search methods using a variety of algorithmic advances or simple heuristics. We briefly describe a few.

The winner of the game, Juha Reunanen, proposed a new variant of cross-validation called cross-indexing, which increases the accuracy of performance prediction in nested cross-validation loops [27]. Closely matching the performances of the winner, Hugo Jair Escalante used a search technique biologically inspired called "particule swarm model selection" [28]. In this method, each candidate model is represented as a particle in the solution space; and by using a population of particles, as well as a fitness function, it emulate the behavior of biological societies (swarm), which objective is to obtain common goals for the entire population. Examples of this behavior on biological populations are bird flocking and fish schooling. Also noteworthy is the method of Gavin Cawley [29] who won the "performance prediction challenge" last year and whose results have not been outperformed in the game. He proposed the use of a Bayesian regularization at the second level of inference, adding a regularization term to the model selection criterion corresponding to a prior over the hyper-parameter values, where the additional regularization parameters are integrated out analytically. Finally, new promising methods of multi-level optimization [30] were proposed at the NIPS workshop where the results of the game were discussed, but must be optimized before they can be applied to sizeable datasets like the ones of the challenge.

The problem of selecting an optimum K in K-fold cross-validation has not been addressed. K=10 seems to be the default value everyone uses.

V. RESULTS

The analysis of the challenge will be added to this paper at revision time, after the challenge terminates on March 1^{st} 2007. It will include a survey of the methods used and conclusions on the effectiveness of agnostic and prior knowledge approaches.

VI. CONCLUSION

This paper presented the design of the IJCNN 2006 competition whose goal is to compare two approaches to machine learning: the "agnostic learning" approach putting all the effort on the classifier and the "prior knowledge" approach capitalizing on human domain knowledge. The challenge was very successful in attracting a large number of participants who competed in the two tracks. The final results will be known on March 1^st 2007. Future work includes incorporating the best identified methods in our challenge toolkit CLOP. We might run a second round of this competition on fresh datasets or a competition on a different theme for IJCNN 2008, depending on the interests of future participants.

ACKNOWLEDGMENTS

We are very thankful to the institutions who originally gave the data. The organization of this challenge was a team effort to which many have participated. We are particularly grateful to Olivier Guyon (MisterP.net) who implemented the back-end of the web site. The front-end follows the design of Steve Gunn (University of Southampton), formerly

used for the NIPS 2003 feature selection challenge. The kind support of Prof. Joachim Buhmann (ETH Zurich) who hosted the competition is gratefully acknowledged. We are thankful to Thomas Fuchs (ETH Zurich) for administering the computer resources. Other advisors and beta-testers are gratefully acknowledged: Kari Torkkola (Motorola, USA), Lambert Schomaker (University of Groningen), Horst Bischof (Graz University, Austria), Fran coise Fogelman (KXEN, Paris, France), Michèle Sebag (LRI, Orsay, France), André Elisseeff (IBM, Zurich, Switzerland), Kristin Bennett (Rensselaer Polytechnic Institute, USA), Gökhan BakIr (MPI for Biological Cybernetics, Germany), Joaquin Quiĕonero Candela (MPI for Biological Cybernetics, Germany), Yi Lu Murphey (University of Michigan-Dearborn, USA), Gavin Cawley (University of East Anglia, UK), Masoud Nikravesh (UC Berkeley, USA), Stefan Lessmann (University of Hamburg, Germany), Erinija Pranckeviciene Institute for Biodiagnostics, NRC, Canada), and Leszek Rutkowski (Technical University of Czestochowa, Poland). The Challenge Learning Object Package (CLOP) is based on code to which many people have contributed [26], [31]. This project is supported by the Pascal network of excellence funded by the European Commission and the National Science Foundation under Grant N0. ECS-0424142. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. KXEN, Microsoft, and Unipen provided additional support permitting to grant prizes to the winners. Amir Safari acknowledges the support of the FWF Austrian Joint Research Project Cognitive Vision under projects S9103-N04 and S9104-N04 and the EU FP6-507752 NoE MUSCLE IST project.

REFERENCES

[1] D. A. Sousa, *How the Brain Learns*. Corwin Press, 2006.
[2] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*, ser. Springer series in statistics. New York: Springer, 2001.
[3] B. Jahne, *Digital Image Processing*. Springer, Berlin, 1991.
[4] Y. LeCun, "Generalization and network design strategies," http://yann.lecun.com/exdb/lenet/, Department of Computer Science, University of Toronto, Tech. Rep. CRG-TR-89-4, 1989.
[5] R. Plamondon and S. Srihari, "Online and off-line handwriting recognition: a comprehensive survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63–84, Jan 2000. http://www.ee.bgu.ac.il/~dinstein/stip2002/HandwritingRecognitionSurveyPAMI.pdf.
[6] P. Baldi, *Bioinformatics: The machine learning approach*. Cambridge, Massachusetts: The MIT press, 2001.
[7] R. S. Rainwater, "Robot competition FAQ," http://robots.net/rcfaq.html, 2007.
[8] "DARPA grand chellenge," http://www.darpa.mil/grandchallenge/, 2007.
[9] "RoboCup," http://www.robocup.org/, 2007.
[10] "Text REtrieval Conference (TREC)," http://trec.nist.gov/, 2007.
[11] "Knowledge Discovery in Databases (KDD) cup," http://www.kdnuggets.com/datasets/kddcup.html, 2007.
[12] "Critical Assessment of techniques for protein Structure Prediction (CASP)," http://predictioncenter.org/, 2007.
[13] "International Conference on Document Analysis(ICDAR)," http://www.icdar2007.org/competition.html, 2007.
[14] "The European Community PASCAL network," http://www.pascal-network.org/Challenges/, 2007.
[15] S. F. Crone, "NN3 neural network forecasting competition." http://www.neural-forecasting-competition.com, 2007.
[16] "The Netflix prize," http://www.netflixprize.com/, 2007.
[17] I. Guyon, S. Gunn, A. Ben-Hur, and G. Dror, "Result analysis of the nips 2003 feature selection challenge," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 545–552.
[18] I. Guyon, A. Saffari, G. Dror, and J. Buhmann, "Performance prediction challenge," in *IEEE/INNS conference IJCNN 2006*, Vancouver, July 16-21 2006.
[19] I. Guyon, "Datasets for the agnostic learning *vs.* prior knowledge competition," http://clopinet.com/isabelle/Projects/agnostic/Dataset.pdf, Clopinet, Tech. Rep., 2005.
[20] R. Kohavi and B. Becker, "The Adult database," ftp://ftp.ics.uci.edu/pub/machine-learning-databases/adult/, 1994.
[21] Y. LeCun and C. Cortes, "The MNIST database of handwritten digits," http://yann.lecun.com/exdb/mnist/, 1998.
[22] J. M. C. Associate Director, "The DTP AIDS antiviral screen program," http://dtp.nci.nih.gov/docs/aids/aids_data.html, 1999.
[23] T. Mitchell, "The 20 Newsgroup dataset," http://kdd.ics.uci.edu/databases/20newsgroups/20newsgroups.html, 1999.
[24] J. A. Blackard and D. J. Dean, "Forest cover type," http://kdd.ics.uci.edu/databases/covertype/covertype.html, 1998.
[25] I. Guyon, et al., "Agnostic learning *vs.* prior knowledge challenge FAQ," http://clopinet.com/isabelle/Projects/agnostic/PFAQ.html, 2005.
[26] J. Weston, A. Elisseeff, G. BakIr, and F. Sinz, "The Spider machine learning toolbox," *http://www.kyb.tuebingen.mpg.de/bs/people/spider/*, 2005.
[27] J. Reunanen, "Resubstitution error is useful for guiding feature selection," *Submitted to JMLR*, 2006.
[28] H. J. Escalante, "Particle swarm model selection," *Submitted to JMLR*, 2006.
[29] G. C. Cawley and N. L. C. Talbot, "Preventing over-fitting during model selection using bayesian regularisation," *Submitted to JMLR*, 2006.
[30] K. Bennett, "Tutorial on mathematical programming for multi-level optimization," http://www.clopinet.com/isabelle/Projects/NIPS2006/Bennett.pdf, 2006.
[31] A. Saffari and I. Guyon, "Quick start guide for CLOP," Graz University of Technology and Clopinet, Tech. Rep., May 2006. http://ymer.org/research/files/clop/QuickStartV1.0.pdf.

APPENDIX

THE CHALLENGE LEARNING OBJET PACKAGE (CLOP)

A. Getting started

The CLOP package can be downloaded from the web-site of the challenge (`http://www.agnostic.inf.ethz.ch/models.php`). For more information on CLOP, see the Quick Start manual [31].

B. Learning objects

The Spider package on top of which CLOP is built, uses Matlab objects (The MathWorks, http://www.mathworks.com/). Two simple abstractions are used:

- **data**: Data objects include two members X and Y, X being the input matrix (patterns in lines and features in columns), Y being the target matrix (i.e. one column of +-1 for binary classification problems).
- **algorithms**: Algorithm objects representing learning machines (e.g. neural networks, kernel methods, decision trees) or preprocessors (for feature construction, data normalization or feature selection). They are constructed from a set of hyper-parameters and have at least two methods: train and test. The train method adjusts the parameters of the model. The test method processes data using a trained model.

For example, you can construct a data object `D`:

```
> D = data(X, Y);
```

The resulting object has 2 members: `D.X` and `D.Y`. Models are derived from the class algorithm. They are constructed using a set of hyperparameters provided as a cell array of strings, for instance:

```
> hyperparam = {'h1=val1', 'h2=val2'};
> model0 = algorithm(hyperparam);
```

In this way, hyperparameters can be provided in any order or omitted. Omitted hyperparameters take default values.

To find out about the default values and allowed hyperparameter range, one can use the "default" method:

```
> default(algorithm)
```

The constructed model `model0` can then be trained and tested:

```
> [Dout, model1] = train(model0, Dtrain);
> Dout = test(model1, Dtest);
```

`model1` is a model object identical to `model0`, except that its parameters (some data members) have been updated by training. Matlab uses the convention that the object of a method is passed as first argument as a means to identify which overloaded method to call. Hence, the "correct" `train` method for the class of `model0` will be called. Since Matlab passes all arguments by value, `model0` remains unchanged. By calling the trained and untrained model with the same name, the new model can overwrite the old one. Repeatedly calling the method "train" on the same model may have different effects depending on the model.

To save the model is very simple since Matlab objects know how to save themselves:

```
> save('filename', 'modelname');
```

This feature is very convenient to make results reproducible, particularly in the context of a challenge.

C. Compound models: chains and ensembles

The Spider (with some CLOP extensions) provides ways of building more complex "compound" models from the basic algorithms with two abstractions:

- **chain**: A chain is a learning object (having a train and test method) constructed from an array of learning objects. Each array member takes the output of the previous member and feeds its outputs to the next member.
- **ensemble**: An ensemble is also a learning object constructed from an array of learning objects. The trained learning machine performs a weighted sum of the predictions of the array members. The individual learning machines are all trained from the same input data. The voting weights are set to one by default. An interface is provided for user-defined methods of learning the voting weights.

Compound models behaves like any other learning object: they can be trained and tested. In the following example, a chain object `cm` consists of a feature standardization for preprocessing followed by a neural network:

```
> cm=chain({standardize, neural});
```

While a chain is a "deep" structure of models, an ensemble is a "wide" structure. The following command creates an ensemble model `em`:

TABLE II
CLOP MODULES PROVIDED FOR THE GAME.

Object name	Description
Preprocessing	
`standardize`	Subtract feature mean and divide by stdev.
`normalize`	Divide patterns by their Euclidean norm.
`shift_n_scale`	Offset and scale all values.
`pc_extract`	Construct features from principal components.
`subsample`	Take a subsample of training patterns.
Feature selection	
`s2n`	Signal-to-noise ratio filter method.
`relief`	Relief filter method.
`gs`	Gram-Schmidt orthogonalization forward selection.
`rffs`	Random Forest feature selection.
`svcrfe`	SVC-based recursive feature elimination.
Classifier	
`kridge`	Kernel ridge regression
`naive`	naive Bayes classifier
`gentleboost`	Regularized boosting
`neural`	Two layer neural network
`rf`	Random Forest (ensemble of trees)
`svc`	Support vector classifier
Postprocessing	
`bias`	Post-fitting of the bias value.

```
> em=ensemble({neural, kridge, naive});
```

To create more complex compound models, models of the same class with different hyperparameters or different models can be combined in this way; chains can be part of ensembles or ensembles can be part of chains.

D. Model selection

The Spider provides several objects for cross-validation and other model selection methods. The challenge participants may elect to use those built-in objects or write their own. A model selection object is derived from the class "algorithm" and possesses train and test methods. For instance, 10-fold cross-validation is performed as follows:

```
> cv_model=cv(my_model, {'folds=10'});
> cv_output = train(cv_model, Dtrain);
```

E. Modules provided for the model selection game

Until December 1^{st} 2006, the challenge participants had the opportunity to participate in a model selection game using CLOP. For the purpose of the game, a valid *model* was defined as a combination of learning objects from a predefined list (type `whoisclop` at the MATLAB prompt to get a the full list of allowed CLOP learning objects; to check that a particular object is a valid CLOP object, type `isclop(object)`).

A typical model usually (but not necessarily) consists of the following parts:

1) Preprocessing
2) Feature Selection
3) Classification
4) Postprocessing

In Table E, we show a list of the modules provided.