AlvsPK Challenge: FACT SHEET

Title: ***Mr. (Ph. D. student)***
Name, ***Hugo Jair Escalante***
address, ***Luis Enrique Erro # 1, Tonantzintla, 72840, Puebla, México***
email: **hugojair@ccc.inaoep.mx**
Acronym of your best entry:
Corrida_Final (according the March 1st milestone results)

Reference:

H. Jair Escalante and M. Montes and L. E. Sucar, ***PSMS for Neural Networks on the IJCNN 2007 Agnostic vs Prior Knowledge Challenge***, In INNS-IEEE Proceedings of the 20th International Joint Conference on Neural Networks 2007 (IJCNN-2007), Orlando, FL, USA.

Method:

Summarize the algorithms you used in a way that those skilled in the art should understand what to do. Profile of your methods as follows:

- <u>Preprocessing</u>
  The following preprocessing methods from the CLOP package were considered for the model selection process: *normalization (normalize), standardization (standardize) and scaling (shif_n_scale).*
- <u>Feature selection</u>
  The feature selection methods considered for the model selection process were: Signal-to-noise ratio (s2n), relief (relief), and Gram-Schmidt orthogonalization (gs).
- <u>Classification</u>
  Experiments were performed with a linear classifier, naïve Bayes and neural networks, though the best ranked entries were obtained with neural networks, a non-linear classifier.
  No ensemble methods were considered for the model selection process.
  No methods for learning from unlabeled data were used.
- <u>Model selection/hyperparameter selection</u>
  For model and hyperparameter selection a bio-inspired search algorithm called: particle swarm optimization (*PSO*) was used. *PSO* is a population-based algorithm that aims to simulate the social behavior of birds within a flock. Candidate solutions of an optimization problem are considered particles that fly through the search space. Each particle has a velocity that is influenced by the global best solution and the best solution the particle has found so far. A fitness function is used to evaluate each candidate solution.

  For model and hyperparameter selection CLOP models (preprocessing method + feature selection method + hyperparameters for the neural net) were codified as real valued vectors. The BER value obtained by 5-fold cross validation was used as fitness function. A standard *PSO* algorithm was implemented using default parameter values, see the reference.

The algorithm was run for 100 iterations for the HIVA, GINA and SYLVA datasets, and for 500 iterations for the ADA dataset; though it was not applied to NOVA because of its high dimensionality and the complexity of the learning machine. Instead, for NOVA the model was selected empirically by trial and error.

The *PSO* algorithm has the same computational drawbacks that any search algorithm applied to the task of model selection, namely they depend on the complexity of the learning machine used, which in turn depends on the size and/or dimensionality of the data. However *PSO* has fast convergence to minima, although it can be a local one. Therefore strategies for avoiding local minima should be included. The simplest (and the one we considered) is the insertion of an inertia weight into the updating velocity equation (see reference paper) that allows the algorithm to perform global and local search, avoiding in a sense local minima.

Results:

Table 1: Our methods best results

Dataset	Entry name	Entry ID	Test BER	Test AUC	Score	Track
ADA	Corrida_final_10CV	922	0.1804	0.9015	0.09	Agnos
GINA	AdaBoost	170	0.053	0.9878	0.3846	Agnos
HIVA	Corrida_final	919	0.2854	0.7551	0.0884	Agnos
NOVA	AdaBoost	170	0.0504	0.9895	0.1987	Agnos
SYLVA	PSMS_100_4all_NCV	987	0.0084	0.9989	0.2362	Agnos
Overall	PSMS_100_4all_NCV	987	0.1178	0.925	0.2464	Agnos

Table 2: Winning entries of the AlvsPK challenge

Best results agnostic learning track						
Dataset	**Entrant name**	**Entry name**	**Entry ID**	**Test BER**	**Test AUC**	**Score**
ADA	Roman Lutz	LogitBoost with trees	13, 18	0.166	0.9168	0.002
GINA	Roman Lutz	LogitBoost/Doubleboost	892, 893	0.0339	0.9668	0.2308
HIVA	Vojtech Franc	RBF SVM	734, 933, 934	0.2827	0.7707	0.0763
NOVA	Mehreen Saeed	Submit E final	1038	0.0456	0.9552	0.0385
SYLVA	Roman Lutz	LogitBoost with trees	892	0.0062	0.9938	0.0302
Overall	Roman Lutz	LogitBoost with trees	892	0.1117	0.8892	0.1431
Best results prior knowledge track						
Dataset	**Entrant name**	**Entry name**	**Entry ID**	**Test BER**	**Test AUC**	**Score**
ADA	Marc Boulle	Data Grid	920, 921, 1047	0.1756	0.8464	0.0245
GINA	Vladimir Nikulin	vn2	1023	0.0226	0.9777	0.0385
HIVA	Chloe Azencott	SVM	992	0.2693	0.7643	0.008
NOVA	Jorge Sueiras	Boost mix	915	0.0659	0.9712	0.3974
SYLVA	Roman Lutz	Doubleboost	893	0.0043	0.9957	0.005
Overall	Vladimir Nikulin	vn3	1024	0.1095	0.8949	0.095967

- quantitative advantages (e.g. compact feature subset, simplicity, computational advantages)
  The PSO algorithm for model selection provides competitive models even when the available algorithms to select from are very simple. Such models are obtained by running the algorithm only a few iterations. It can be considered a black-box method in the sense that no experience on machine learning is required to use it.

- qualitative advantages (e.g. compute posterior probabilities, theoretically motivated, has some elements of novelty).
  The PSO algorithm has been already used for training neural network (adjusting weights), though it has not been used for model selection with hyperparameter selection, a slight modification that allows for the selection of preprocessing and feature selection algorithms as well. The algorithm can be used with any other learning algorithm and preprocessing/feature selection methods.

Code: If CLOP or the Spider were used, fill out the table:

Dataset	Spider command used to build the model
ADA	chain({standardize({'center=0'}),normalize({'center=1'}),shift_n_scale({'take_log=0'}),neural({'units=5', 'shrinkage=1.4323','balance=0','maxiter=257'}), bias})
GINA	chain({gs({'f_max=48'}),shift_n_scale({'take_log=1'}),neural({'units=16', 'shrinkage=0.29191','balance=1','maxiter=456'}), bias})
HIVA	chain({standardize({'center=1'}),normalize({'center=0'}),neural({'units=5', 'shrinkage=3.028','balance=0','maxiter=448'}), bias})
NOVA	chain({normalize({'center=0'}),gentleboost(neural({'units=1', 'shrinkage=0.2' , 'balance=1', 'maxiter=50'}) , {'units=10','rejNum=3'}), bias})
SYLVA	chain({standardize({'center=1'}),normalize({'center=1'}),neural({'units=6', 'shrinkage=0.02882','balance=1','maxiter=359'}), bias})

If new Spider functions were written or if CLOP or the Spider were not used, briefly explain your implementation. Provide a URL for the code (if available). Precise whether it is a push-button application that can be run on benchmark data to reproduce the results, or resources such as modules or libraries.

Keywords: Put at *least one keyword in each category*. Try some of the following keywords and add your own:

- Preprocessing or feature construction: centering, scaling, standardization.
- Feature selection approach: embedded feature selection.
- Feature selection engine: correlation coefficient, Relief.
- Feature selection search: feature ranking.
- Feature selection criterion: K-fold cross-validation.
- Classifier: neural networks.
- Hyper-parameter selection: pattern search, bio-inspired search cross-validation, K-fold.
- Other: swarm optimization.