AlvsPK Challenge: FACT SHEET

Title:
Research Associate at Institute of Software Systems, Kiev, Ukraine

Name, address, email:
Dmitry Zhora
40 Glushkov pr., Kiev, 03187, Ukraine
dvz73@bigfoot.com

Acronym of your best entry:

Reference:
Provide a pointer to a longer technical memorandum or to an IJCNN paper (optional).
Classifier description:
D.V. Zhora. Evaluating Performance of Random Subspace Classifier on ELENA Classification Database // Proc. Int. Conf. Artificial Neural Networks 2005, LNCS 3697, pp. 343-349.
Relevant articles:
D.V. Zhora. Financial Forecasting using Random Subspace Classifier // Proc. Int. Joint Conf. Neural Networks 2004, vol. 4, pp. 2735-2740.
D.V. Zhora. Data Preprocessing for Stock Market Forecasting using Random Sub-space Classifier Network // Proc. Int. Joint Conf. Neural Networks 2005, pp. 2549-2554.
D.V. Zhora. Analysis of separating surfaces formed by a random subspace classifier. // Cybernetics and Systems Analysis, Springer, Vol. 42, Num. 6, Nov. 2006, pp. 817-830.
D.V. Zhora. Analysis of a Classifier with Random Thresholds. // Cybernetics and Systems Analysis, Springer, Vol. 39, Num. 3, May 2003, pp. 379-393.

Some information is available at http://rsc.netfirms.com/rsclass/index.htm.

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:
Random subspace classifier is a high-performance neural network classifier, which can provide the solution for complex multidimensional and overlapping class distributions. It's quite competitive when the number of input parameters and training set size increase. The classifier consists of two parts: the first part makes a nonlinear transformation of a input real vector into a high-dimensional binary vector, presented by the hidden layer; the second part of the classifier is a one-layer perceptron. The classifier uses a coarse coding technique to transform the input vector into the binary representation. Thus, class representatives are likely to become linearly separable. The classifier can be considered as a discrete counterpart of the RBF network, the difference is that all operations are discrete and the shape of the hidden layer neuron activation function is not radial. Another consideration is that RSC is similar to the SVM. In this case both approaches use nonlinear transformation of the input vector into the high-dimensional feature space. In contrast to the SVM, RSC does specify the type of the transformation, but doesn't use

optimization technique to provide "good" linear separation surface for reasons of computational efficiency. At the same time, the RSC can implement the decision rule obtained using another linear learning machine.

- Preprocessing
  No special preprocessing was done to the datasets. However, internally the classifier linearly maps each vector component to the range [0,1].
- Feature selection
  No feature selection procedures were made (unfortunately).
- Classification
  - What engine did you use? (Precise whether the classifiers used are linear. For kernel methods, indicate what kernel is used.)
    The vectors were transformed to "hidden-layer" space using kernel
    Bj = 1, if $\forall i \in 1,\ldots,h: \; l_{ij} < x_{j\,(i,j)} < h_{ij}$
    Bj = 0 otherwise.
    See referenced articles for details. The classification is linear in the "hidden-layer" space.
  - Did you use ensemble methods?
    No (unfortunately)
  - Did you use "transduction" or learning from the unlabeled test set?
    Transduction approaches were not used, test set wasn't used as well. However, there is the possibility to estimate probability distribution more accurately (without class information) using unlabeled test data.
- Model selection/hyperparameter selection

Random subspace classifier hyperparameters:

1. Distance between corresponding thresholds – always 1, other values were not tested.
2. Hiddden layer size – typically 32768, 65536 for Silva.
3. Subspace dimension – always 3, other values were not tested.
4. Whether to use "sensitive structure" when the density of thresholds is proportional to the density of data points.
5. Whether to use error correction or "stochastic approximation" learning procedure. Error correction was always used (very simple rule suggested by Rosenblatt for one-layer perceptron).
6. Whether to conduct full training (until the training set is interpreted without errors, good for low error tasks) or "save-best" training (to stop early in the case of high error tasks). Different choices.
7. The number of epochs for save-best alorithm. Different numbers.

Results:

Table 1: Our methods best results

Dataset	Entry name	Entry ID	Test BER	Test AUC	Score	Track
ADA	rsc.ss.ec.sb.ber	970	0.2292	0.7703	0.8466	Agnos
GINA	rsc.ec	954	0.0855	0.915	0.6496	Agnos
HIVA	rsc.ss.ec.sb.ber	1018	0.3149	0.6888	0.6305	Agnos
NOVA	nova2.rsc.ec.sb.ber	1058	0.0692	0.932	0.4423	Agnos
SYLVA	rsc.ec.ber	942	0.4894	0.5106	0.9899	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 classifier is relatively fast, the only floating point operation used is comparison, all other operations are discrete (integer, logical etc.).
- qualitative advantages (e.g. compute posterior probabilities, theoretically motivated, has some elements of novelty).
  The classifier is very competitive in the case of complex multidimentional and low-Bayes error tasks. Uses SVM (and RBF) network architecture.

Code: If CLOP or the Spider were used, fill out the table:
CLOP or Spider were not used.

Dataset	Spider command used to build the model
ADA	
GINA	
HIVA	
NOVA	
SYLVA	

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: standardization
- Feature selection approach:
- Feature selection engine:
- Feature selection search: stochastic search relatively to correlation coefficient between hidden neuron output and the class label is applicable but not used (unfortunately)
- Feature selection criterion:
- Classifier: neural network, kernel-method
- Hyper-parameter selection: cross-validation
- Other: coarse coding